JULY 1992

COMPUTER

TRANSCRIBING FOR REUSE

Document Image Analysis Systems

PRE-REGISTER FOR THE
17th Annual Conference on Local Computer Networks

September 13-16, 1992
Minneapolis Metrodome Hilton– Minneapolis, Minnesota USA
Sponsored by the IEEE Computer Society

Tutorials • Technical Paper Sessions • Expert Panel Discussions • Workshop-style Interchange

Registration Information: Pre-registration must be received on or before **Monday, September 6, 1992**–no exceptions. The fees for the tutorial on Sunday, September 13 include tutorial notes and one break. The fees for the tutorials on Monday, September 14 include luncheon, tutorial notes, and two breaks. The fees for the conference on September 15 and 16 include a copy of the proceedings, four breaks, two luncheons, and the banquet. Make all checks payable to: *17th Conference on LCN*

For advanced program and hotel information contact: LANWORKS
(612) 591-5837 / (612) 591-7772-Fax

Hotel: A block of rooms has been reserved at the conference rate ($75) for conference attendees. Reservations must be made by August 14 to guarantee room availability and conference rate. Contact:

Minneapolis Metrodome Hilton–
1330 Industrial Blvd., Minneapolis, MN 55413
USA (612) 331-1900, (612) 331-6827-Fax
Rates: $75 (single or double)
Reference: *Local Computer Networks Conference*

TUTORIALS

Sunday, September 13 (1:00 p.m. - 5:00 p.m.)	Tutorial 1	*The Xpress Transport Protocol (XTP)*	Alfred Weaver, University of Virginia
	Tutorial 2	*Multimedia Networks*	William Seifert, Multimedia Networks
Monday, September 14 (9:00 p.m. - 5:00 p.m.)	Tutorial 3	*FDDI*	Floyd Ross, Ascom Timplex
	Tutorial 4	*Understanding SNMP*	Marshall Rose, Dover Beach Consulting
	Tutorial 5	*Design/Application of ATM Networks*	Jonathan Turner, Washington University

TECHNICAL PROGRAM

Tuesday, September 15	Expert Panel Track	Keynote Address Technical Paper Track I Conference Banquet	Technical Paper Track II
Wednesday, September 16	Expert Panel Track	Technical Paper Track I	Technical Paper Track II

Send completed form & fee (*payable in U.S. Dollars*) to:
17th LCN Conference
Harvey A. Freeman
LANWORKS, Inc.
6465 Wayzata Blvd. Suite 115
St. Louis Park, MN 55426
USA (612) 591-5837

Registration Fees (Circle Tutorial Number)	Conference	Tutorial 1 or 2	Tutorial 3, 4, or 5
Students: (*Include copy of paid fee statement*)	$200_____	$125_____	$250_____
IEEE Members:	$330_____	$125_____	$250_____
Non-Members:	$415_____	$160_____	$320_____

Name _____

Organization _____

Street _____

City, State, Zip _____

Phone _____

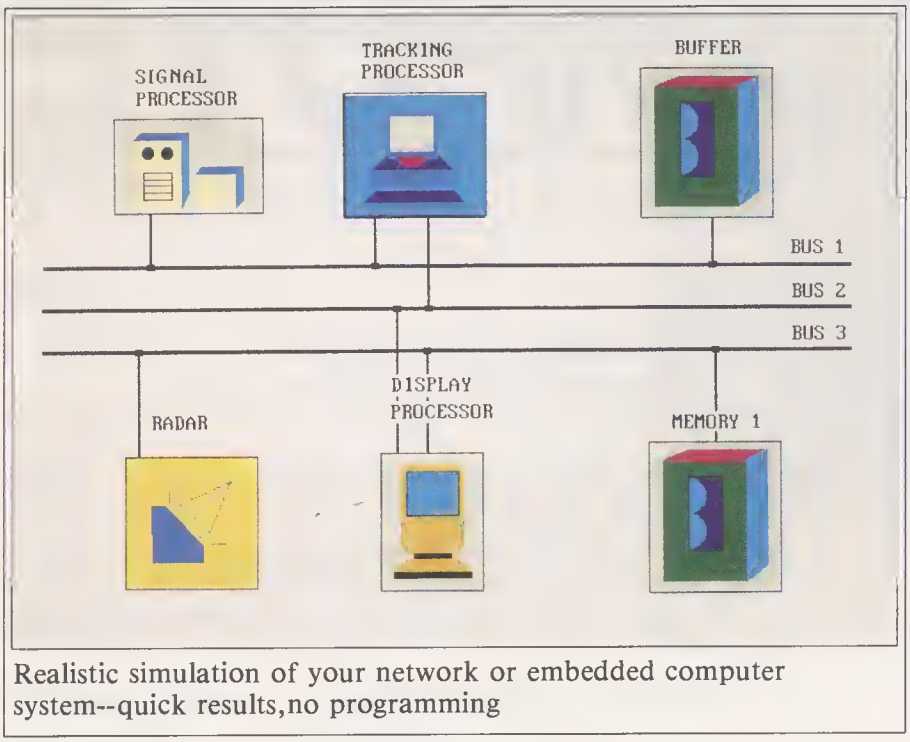
Registrations received after Sept. 6 will be considered late. Late and on-site registrations will be at the following rates:

	Conf.	Tut. 1-2	Tut. 3-5
IEEE Member:	$395	$150	$300
Non-Member:	$495	$195	$385

REGISTER EARLY

CP

NEW FOR NETWORK ANALYSTS AND SYSTEMS ENGINEERS



Realistic simulation of your network or embedded computer system--quick results,no programming

Simulation Conference 17
August 25-27, 1992
Washington, D.C.
Call Art at (619) 457-9681
for details

NETWORK II.5 now predicts performance of computer-communication systems

Free trial and, if you act now, free training

NETWORK II.5 uses simulation to predict your network performance. You simply describe your network and workload.

Animated simulation follows immediately--no programming delays.

Easy-to-understand results

You get an animated picture of your network. System bottlenecks and changing levels of utilization are apparent.

Seeing your network animated increases everyone's understanding of its operation and builds confidence in your results.

Your reports show response times, messages delivered, messages lost, device utilization, and queueing statistics.

Computers with NETWORK II.5

NETWORK II.5 is available for most PC's, Workstations, and Mainframes.

Your network simulated

You can analyze embedded or distributed computer systems, or other computer-communication networks. Industry standard protocols such as FDDI and IEEE Standard 802.X are built-in. Others can be modeled.

You can easily study the effect of changing network parameters or even network protocols.

You can simulate some portions of the network at a detailed level and others at a coarser level.

Free trial information

The free trial contains everything you need to try NETWORK II.5® on your computer. For a limited time we also include free training **--no cost, no obligation.**

Call Paul Gorman at (619) 457-9681, Fax (619) 457-1184. In Europe, call Nigel McNamara, in the UK, on 0276 671 671, Fax 0276 670 677. In Canada, call Peter Holt on (613) 782-2474, Fax (613) 782-2202.

Free trial offer

☐**Yes I want to see how NETWORK II.5 quickly answers network performance questions.**

Limited offer--Act now for free training.

Name

Organization

Address

City State Zip

Telephone Fax

Computer Op. Syst.

☐Send details on your University Offer.

Return to:
CACI Products Company
3344 North Torrey Pines Court
La Jolla, California 92037
Call Paul Gorman at (619) 457-9681
Fax (619) 457-1184

Mie comp

In Europe:
CACI Products Division
Coliseum Business Centre
Watchmoor Park, Riverside Way
Camberley, Surrey GU15 3YL, UK
Call Nigel McNamara on 0276 671 671
Fax 0276 670 677

In Canada:
CACI Products Company
200-440 Laurier Avenue West
Ottawa, Ontario K1R 7X6
Call Peter Holt on (613) 782-2474
Fax (613) 782-2202

NETWORK II.5 is a registered trademark and service mark of CACI Products Company.
©1992 CACI Products Company.

COMPUTER

July 1992 • Vol. 25, No. 7

ARTICLES

5 **Guest Editors' Introduction: Document Image Analysis Systems**
Lawrence O'Gorman and Rangachar Kasturi

10 **A Prototype Document Image Analysis System for Technical Journals**
George Nagy, Sharad Seth, and Mahesh Viswanathan
Intelligent document segmentation can bring electronic browsing within the reach of most users. The authors show how this is achieved through document processing, analysis, and parsing the graphic sentence.

25 **An Interpretation System for Land Register Maps**
Luca Boatto, Vincenzo Consorti, Monica Del Buono, Silvano Di Zenzo, Vincenzo Eramo, Alessandra Esposito, Francesco Melcarne, Marco Meucci, Andrea Morelli, Marco Mosciatti, Stefano Scarci, and Marco Tucci
The semantics of land register maps drive this document conversion system. However, its methods of image representation, vectorization, and symbol recognition can be generalized to other classes of line drawings.

34 **Postal Address Block Location in Real Time**
Paul W. Palumbo, Sargur N. Srihari, Jung Soh, Ramalingam Sridhar, and Victor Demjanenko
A postal automation system locates destination address blocks on letter mail pieces with a high success rate. Pipelining and multiprocessor techniques achieve real-time processing speed.

46 **Celesstin: CAD Conversion of Mechanical Drawings**
Pascal Vaxivière and Karl Tombre
A prototype CAD conversion system extracts higher level structures for knowledge-based analysis. It recognizes such entities as screws, ball bearings, and shafts.

PROJECT OVERVIEWS

56 **A Robust Recognition System for a Drawing Superimposed on a Map**
Shigeyoshi Shimotsuji, Osamu Hori, Mieko Asano, Kaoru Suzuki, Funihiko Hoshino, and Toshiaki Ishii

59 **Reading Handwritten Digits: A ZIP Code Recognition System**
Ofer Matan, Henry S. Baird, Jane Bromley, Christopher J.C. Burges, John S. Denker, Lawrence D. Jackel, Yann Le Cun, Edwin P.D. Pednault, William D. Satterfield, Charles E. Stenard, and Timothy J. Thompson

63 **From Paper to Office Document Standard Representation**
Andreas Dengel, Rainer Bleisinger, Rainer Hoch, Frank Fein, and Frank Hönes

67 **DRS: A Workstation-Based Document Recognition System for Text Entry**
Tomio Amano, Akio Yamashita, Nobuyasu Itoh, Yoshinao Kobayashi, Shin Katoh, Kazuharu Toyokawa, and Hiroyasu Takahashi

71 **Off-Line Arabic Character Recognition**
Habib Goraine, Mike Usher, and Samir Al-Emami

75 **Understanding Diagrams in Technical Documents**
Robert P. Futrelle, Ioannis A. Kakadiaris, Jeff Alexander, Catherine M. Carriero, Niko Nikolakis, and Joseph M. Futrelle

Published by the IEEE Computer Society

Circulation: *Computer* (ISSN 0018-9162) is published monthly by the IEEE Computer Society, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264; phone (714) 821-8380. IEEE Computer Society Headquarters, 1730 Massachusetts Ave. NW, Washington, DC 20036-1903; IEEE Headquarters, 345 East 47th St., New York, NY 10017. Annual subscription included in society member dues. Nonmember subscription rates: available upon request. Single copy prices: members $10.00; nonmembers $20.00. This magazine is also available in microfiche form.

Postmaster: Send undelivered copies and address changes to *Computer*, IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08855. Second class postage is paid at New York, New York, and at additional mailing offices. Canadian GST #125634188. Third class mail enclosed.

Bulk Rate
U.S. Postage
PAID
Shepherdsville, KY
Permit No. 93

Copyright and reprint permission: Copyright © 1992 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US copyright law for private use of patrons: (1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress Street, Salem, MA 01970; (2) pre-1978 articles without fee. Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint, or republication permission, write to Permissions Editor, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264.

Editorial: Unless otherwise stated, bylined articles, as well as product and service descriptions, reflect the author's or firm's opinion. Inclusion in *Computer* does not necessarily constitute endorsement by the IEEE or the Computer Society. All submissions are subject to editing for style, clarity, and space.

BPA

DEPARTMENTS

81 **New Products**

84 **Product Reviews**
CASE on the Mac

91 **IC Announcements**

92 **Microsystem Announcements**

93 **Computer Society News**
Fall election slate, 1992 Eckert-Mauchly Award, global networking

97 **Update**

99 **Standards**

100 **Call for Papers**

102 **Calendar**

110 **Book Reviews**

112 **Open Channel**

Monks copying manuscripts in a medieval scriptorium reproduced text, graphics, and pictures in separate operations, just as today's document image analysis systems identify and extract these elements for later reuse. First, a scribe copied sections of text, which a second monk proofread. Next, the sheets went to the rubricator, who inserted titles, headlines, initials, notes, and the like. Finally, the illuminator added illustrations, and the volume was ready for binding.

Cover: Alexander Torres

Career Opportunities, 106; Computer Society Information, 24; Membership Application, 79; Change of Address Form, 74; Advertiser/Product Index, 80; Reader Service Card, 80A

In the next issue

Multitopic issue including articles on parallel programming, visual feedback control, Flat Concurrent Prolog, flexible text display, and demonstrational interfaces

COMPUTER

10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264

Editor-in-Chief
Jon T. Butler, Naval Postgraduate School
Associate Technical Editor
David C. Rine, George Mason University
Editorial Board
Dennis R. Allison, Stanford University
Laxmi N. Bhuyan, Texas A&M University
Thomas A. Corbi, IBM
Scott Davidson, AT&T
Richard Eckhouse, MOCO
Michael Evangelist, Andersen Consulting
Scott E. Fahlman, Carnegie Mellon University
John R. Gurd, University of Manchester
Brent T. Hailpern, IBM T.J. Watson Research Center
Tadao Ichikawa, Hiroshima University
Jogikal Jagadeesh, Ohio State University
Alan Kaminsky, Rochester Institute of Technology
Yale Patt, University of Michigan
Guylaine M. Pollock, Sandia National Laboratories
Norman F. Schneidewind, Naval Postgraduate School
Vincent Y. Shen, Hong Kong University of Science and Technology
Martha Sloan, Michigan Technological University
Will Tracz, IBM
Ronald D. Williams, University of Virginia, Charlottesville
Akinori Yonezawa, University of Tokyo

Submissions
Send 12 double-spaced copies of articles and special-issue proposals to Jon T. Butler, Dept. of Electrical and Computer Engineering, Code EC/Bu, Naval Postgraduate School, Monterey, CA 93943-5004, phone (408) 646-3299. For electronic submission, Butler's addresses are j.butler on Compmail II and butler@ece.nps.navy.mil on Internet.

Staff
Managing Editor: Marilyn Potes
Staff Editors: Christine Miller, Chuck Governale, Bob Carlson, Linda World
Contributing Editor: Ware Myers
Art Consultant: Jay Simpson
Design: Toni Van Buskirk
Production: Chris Patterson, Ken Duckworth
Publisher: True Seaborn
Editorial Director, CS Magazines: Marilyn Potes
Assistant to the Publisher: Pat Paulsen
Membership/Circulation Manager: Christina Champion
Advertising Manager: Heidi Rex
Advertising Coordinator: Marian Tibayan

Magazine Operations Committee
James J. Farrell III (chair), Valdis Berzins, Jon T. Butler, B. Chandrasekaran, Carl Chang, Manuel d'Abreu, Dante Del Corso, John A.N. Lee, Peter R. Wilson

Publications Board
Harold Stone (chair), James J. Farrell III, Ronald G. Hoelzeman, Mary Jane Irwin, Ming T. (Mike) Liu, Michael C. Mulder, Theo Pavlidis, Sallie V. Sheppard, Kishor Trivedi

FRONTIERS '92

THE 4TH SYMPOSIUM ON THE

Frontiers of Massively Parallel Computation

October 19-21, 1992 • McLean Hilton • McLean, Virginia

This symposium is the fourth in a series of biannual meetings on massively parallel computation, focusing on research related to, or adaptable for, systems scalable to 1,000 or more processors. Original research papers about aspects of the design, analysis, development, and/or use of massively parallel computers will be presented, including papers relating to high performance computing and communications. Below are the highlights. There will be more panels, sessions, and social events.

CONFERENCE HIGHLIGHTS

Tue 8:30-10:00 am	***The Federal HPCC Program: Current Expectations From Key Federal Agencies*** *Keynote Panel Chair - Milt Halem, NASA Goddard Space Flight Center*
Tue 6:00- 7:30 pm	***Poster Session and Reception***
Tue 7:30-10:00 pm	***Banquet Speaker: Gordon Bell What Price Teraflops?***
Wed 4:00- 6:00 pm	***Perspectives on Massively Parallel Computation*** *Algorithms - **Leslie Valiant**, Harvard University* *Applications - **Paul Woodward**, Army High Performance Computing Research Center & U Minnesota* *Architectures - **Mike Flynn**, Stanford University,* *System Software - **Lionel Ni**, Michigan State University*
Wed 6:00- pm	***Closing Reception***

General Chair - Pearl Wang
George Mason Univ
Program Chair - H.J. Siegel, Purdue

Program Vice-Chairs
ALGORITHMS Leah Jamieson, Purdue
ARCHITECTURES Ken Batcher,
Kent State Univ
APPLICATIONS Jack Dongarra,
Univ Tenn/ORNL
SOFTWARE Andre van Tilborg, ONR

For a more complete program, write
James Fischer, Frontiers '92
Code 932.1, NASA Goddard,
Greenbelt, MD 20771
e-mail : f92info@cs.gmu.edu

TUTORIALS *MONDAY OCT 19:*

A. *Lennart Johnsson*, Harvard University. ***"Scientific Software for Massively Parallel Architectures"***
B. *Tom Casavant*, U of Iowa. ***"Software Tools for Visualization of Massively Parallel Programs and Systems"***

WORKSHOPS *MONDAY OCT 19:*

"New Frontiers, or Future Directions in Massively Parallel Processing" The workshop's objective is to identify and define new research directions in massively parallel processing. Four Working Groups will focus on the following important themes: *Processor Design, Interconnection Networks, Languages & Compilers,* and *Software Development Tools. Note:* For participation information, and an application, contact Prof. Isaac Scherson, Department of Information & Computer Science, Univ. of California, Irvine, CA 92717, e-mail: wsf92@ics.uci.edu, FAX (714) 856-4056.

COMMERCIAL EXHIBITORS *TUE 8:00-7:00 & WED 8:00-5:00:*

Convex Corporation, Intel Supercomputer Systems Division, Loral Defense Systems, MasPar Computer Corporation, Thinking Machines Corporation, Wavetracer

Credit Card Payments:
Please debit my credit card

Type:_____
For $ _____

Holder's Name:_____
Expir Date: _____
Today's Date:_____

Card Number:
_____
Signature:
_____

REGISTRATION FORM (DISCOUNT FOR IEEE MEMBERS)

Name_____e-mail_____Institution_____
Address_____City_____State_____Zip_____IEEE Member #_____
Phone (w)_____(h)_____ Additional Banquet Tickets #____x $40 = $_____

	Prior to Sept. 25			After Sept. 25			
	Student	Member	Non.	Student	Member	Non.	TOTAL
Conference Reg	$65	$250	$315	$65	$300	$375	_____
Tutorial (each-circle choices) A B		$150	$190		$180	$225	_____

Mail to:- Frontiers '92 (credit card users use supplied form) FEE ENCLOSED $_____
IEEE Computer Society (Please make checks payable to Frontiers '92)
1730 Massachusetts Avenue NW Phone: 202-371-1013
Washington DC 20036 FAX: 202-728-0884

Written requests for refunds must be received no later than Sept. 25. Refunds are subject to a $50 processing fee. All no-shows will be billed in full. Registrations after October 2 will be accepted onsite only.

Credit Card Payments:
Please debit my credit card

Type:_____
For $ _____

Holder's Name:_____

Expir Date: _____
Today's Date:_____

Card Number:
_____
Signature:
_____

LOCATED IN WASHINGTON DC AREA: MCLEAN HILTON IN TYSONS CORNER

Held in the suburbs of Washington DC, with access to national museums, government funding agencies, and international performance arts. Access to Amtrack, or National and Dulles airports.

Name_____Sharing With _____Institution _____
Address_____City_____State_____Zip_____
Phone (w)_____(h)_____Check in _____/10/'92, Time _____Check out_____/10/'92
Double or Single _____ room(s) .#_____@ $105/night. Number of people in room _____
King or Double/Double Bed ____ Smoking/Non Smoking____Handicapped____ FEE ENCLOSED $_____
Mail to:- THE MCLEAN HILTON (credit card users use supplied form)
7620 Jones Branch Drive, Reservations: (703) 761-5111
McLean, Virginia 22102-3308, USA FAX: (703) 761-5100

Rooms will be held until 6PM unless guaranteed with an accepted credit card. Check-in time is after 3PM, Check-out is noon. Room charges subject to a 6.5% tax. Cancellations made after 6PM of check-in day will be billed one night's room and tax. Government rates are available.
CUT OFF DATE: SEPT 21, '92
Reservations made after 9/21 based upon room and rate availability.

With support from *Cray Research, Inc, Intel Supercomputer Systems Division, MasPar Computer Corporation, Thinking Machines Corporation, Virginia Center for Innovative Technology*

NASA Goddard Space Flight Center

IEEE Computer Society

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC
IEEE

Guest Editors' Introduction:

Document Image Analysis Systems

Lawrence O'Gorman, AT&T Bell Laboratories

Rangachar Kasturi, Pennsylvania State University

Who hasn't encountered some of the problems associated with paper documents? For instance, you would really like to excerpt a section from a magazine or business letter but refrain because of the labor of retyping. Or you have a diagram that would be perfect to insert in a document — except for one small but necessary change. Perhaps the problem is the sheer volume of paper or the fact that the sole copy of a document may be lent or lost.

On a larger scale, libraries face the ongoing degradation of paper material. Manufacturing firms have the costly task of maintaining, locating, and modifying their engineering drawings. And the volume of paper traveling through the postal systems is daunting.

Fortunately, researchers in the field of document image analysis are working to solve these problems. This issue describes a range of systems and applications that are part of this effort.

Traditionally, information has been stored and transmitted as paper documents. In the past few decades, documents have increasingly originated on the computer. However, it is unclear whether the computer has decreased or increased the amount of paper. Documents are still printed on paper for reading, dissemination, and markup. The oft-repeated goal in the early 1980s of the "paperless office" has now given way to a different objective: handling the flow of electronic and paper documents in an efficient and integrated way.

0018-9162/92/0700-0005$03.00 © 1992 IEEE

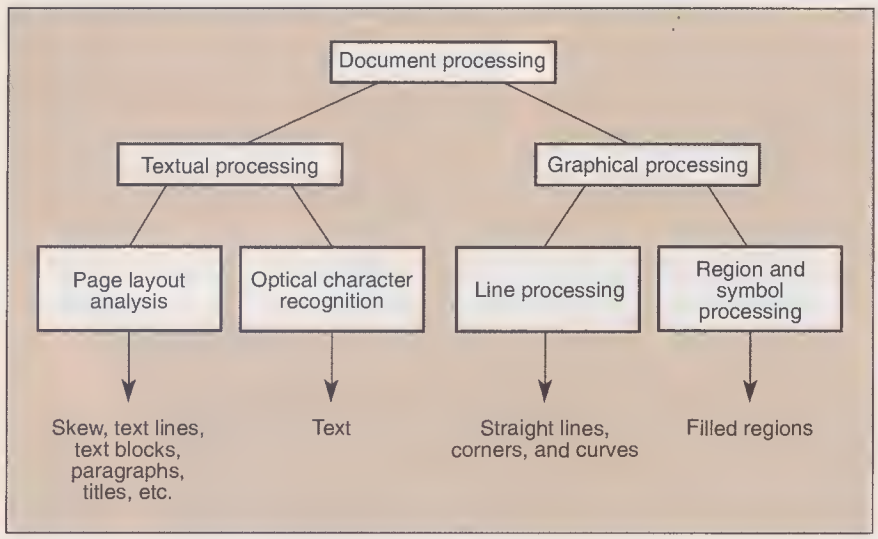


Figure 1. A hierarchy of document processing subareas listing the types of document components dealt with in each subarea.

The ultimate solution would be to deal with paper documents as we deal with other forms of computer media. Paper would remain a comfortable medium for reading and modifying, but it would also be compatible with the computer.

The objective of document image analysis is to recognize text, graphics, and pictures in images and to extract the intended information as a human would. Textual processing and graphical processing are two categories of document image analysis dealing, respectively, with the text components and the graphics components of a document image (see Figure 1). Textual processing includes

- determining the skew (any tilt at which the document may have been scanned);
- finding columns, paragraphs, text lines, and words; and
- performing optical character recognition (OCR).

Graphical processing deals with components such as lines and symbols; therefore, graphics processing includes line thinning, line fitting, corner and curve detection, and symbol recognition. Continuous-tone pictures are a third major document component. Except for recognizing their location on a page, analysis of these components usually falls within the domain of image processing and machine vision. After these techniques are applied, document image analysis systems can cull the megabytes of picture data to produce a much more concise semantic description.

Too many documents in too many paper and electronic formats

It isn't hard to find problems for document analysis to solve, or systems designed to address the problems. Look at the stacks of paper documents around the workplace. Some may be computer generated — inevitably by different computers and software. Some include both formatted text and tables as well as handwritten entries. The documents come in different sizes, from 3.5 × 2-inch business cards to 34 × 44-inch engineering drawings. Many businesses use imaging systems to store "pictures" of the paper and to make retrieval more efficient. Future document analysis systems will recognize types of documents, enable the extraction of logical parts, and translate them from one computer-generated format to another.

The post office is another example. Glance behind the counter at the mounds of letters and packages. Some US post offices handle more than a million pieces of mail each day. Machines have been used for several decades to recognize addresses and sort mail, but the need to process greater volumes faster and more accurately has outpaced existing technologies.

And, finally, there are libraries, with row after row of paper information. Loss of material, misfiling, limited numbers of copies, and even degradation of materials are common problems that document analysis techniques may help to solve.

These examples serve as applications for some of the systems described in this issue.

From pixels to paragraphs and drawings

Figure 2 illustrates a common sequence of steps in document image analysis, and we describe the steps individually.

Data capture and preprocessing. Optical scanning is the usual method of capturing data from a paper document. The scanned data is stored in a file of picture elements, called pixels, that are sampled in a grid pattern from each page of the document. These pixels have values: 0 or 1 for binary images, 0 to 255 for gray-scale images, and three channels of 0 to 255 color values for color images. At a typical sampling resolution of 300 dots per inch, an 8.5 × 11-inch page would yield an image of 2,550 × 3,300 pixels. It is important to understand that a document's image contains only raw data that must be further analyzed to glean information. For instance, the letter *a* printed on an area one-tenth of an inch square is captured as a 30 × 30-pixel matrix of 1 or 0 values. Although humans see this image as the letter *a*, to a computer it is initially just 900 binary values with no more meaning than 900 adjacent bits from any computer memory.

Preprocessing includes binarization, noise reduction, and signal enhance-

ment. For gray-scale images with information that is inherently binary, such as text or graphics, binarization is usually performed first. The objective is to automatically choose a threshold that separates the foreground and background information. An example that has been well researched is the extraction of handwriting from a bank check — especially a bank check containing a scenic image as a background.

Document image noise occurs from image transmission, photocopying, or document degradation due to aging. Salt-and-pepper noise (also called impulse noise, speckle noise, or dirt) is common in binary images. It consists of randomly distributed black specks on white background and white specks on black background. It is reduced by filtering the image; the background is "grown" into the noise specks, thus filling these holes.

Signal enhancement is similar to noise reduction but uses domain knowledge to reconstitute expected parts of the signal that have been lost. An example of signal enhancement is filling gaps and making line widths consistent in graphics corrupted by noise.

Segmentation. There are two levels of segmentation. First, text and graphics are separated for subsequent processing by different methods. Second, text is segmented by locating columns, paragraphs, words, titles, and captions; graphics, too, are segmented, usually by separating symbol and line components. For instance, on a page containing a flowchart with an accompanying caption, text and graphics are first separated. Then the text is separated into that of the caption and that of the chart. Graphical elements are separated into rectangles, circles, connecting lines, etc.

Feature extraction. In a text image, global features describe each page. These consist of skew (the tilt at which the page has been scanned), line lengths, line spacing, etc. In addition, individual characters have local features such as font size, number of loops in a character, number of crossings, accompanying dots, etc., that are used in optical character recognition.

In a graphical image, global features describe the skew of the page, line widths, range of curvature, minimum line lengths, etc. Local features describe each corner, curve, and straight line, as well

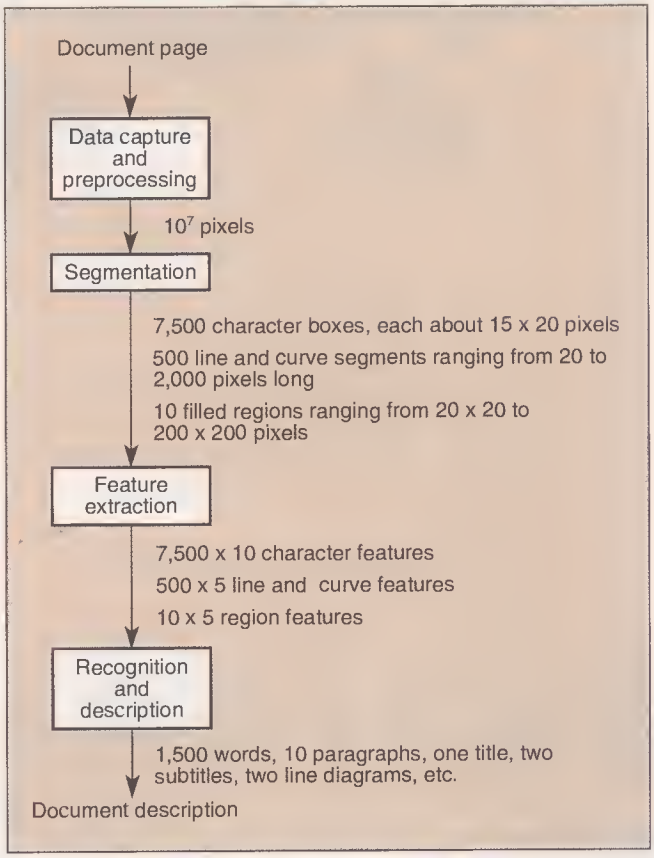


Figure 2. A typical sequence of steps for document analysis, along with examples of intermediate and final results and estimates of data size.

Hardware advancements and the evolution of document image analysis

The field of document analysis can be traced back through a computer lineage that includes digital signal processing and digital image processing. Digital signal processing, initially fostered by the introduction of fast computers and algorithms such as the fast Fourier transform in the mid-1960s, has as its objective the interpretation of one-dimensional signals such as speech and other audio signals. In the early 1970s, with larger computer memories and still faster processors, image processing methods and systems were developed for analyzing two-dimensional signals, including digitized pictures. Specialized fields of image processing are associated with particular applications — for example, biomedical image processing for medical images, machine vision for processing of pictures in industrial manufacturing, and computer vision for processing images of three-dimensional scenes used in robotics.

In the mid to late 1980s, document image analysis began to grow rapidly. Again, this was predominantly due to hardware advances that supported fast processing at reasonable costs. Whereas a typical speech signal is 256 samples per frame and a machine vision image is 512 × 512 pixels, a document image ranges from 2,550 × 3,300 pixels for an 8.5 × 11-inch business page digitized at 300 dots per inch, to 34,000 × 44,000 pixels for a 34 × 44-inch (E-sized) engineering diagram digitized at 1,000 dpi.

Commercial document analysis systems are now available for storing business forms, performing optical character recognition on typewritten text, and compressing engineering drawings. Document analysis research continues to pursue more intelligent handling of documents, better compression — especially through component recognition — and faster processing.

as rectangles, circles, and other geometric shapes.

Recognition and high-level description. In the final step, the components are assigned semantic labels and the entire document is described as a whole. This is the stage that uses domain knowledge most extensively. The result is a document description such as a human would give. For a text image, we refer not to pixel groups or blobs of black on white but to titles, subtitles, bodies of text, footnotes, etc. Depending on the arrangement of these text blocks, a page of text may be a title page of a paper, a table of contents for a journal, a business form, or the face of a piece of mail. For a graphical image — an electrical circuit diagram, for instance — we refer not to lines joining circles, triangles, and other shapes but to connections between AND gates, transistors, and other electronic components. The components and their connections describe a particular circuit that has a purpose in the known domain. This end product enables us to store the document efficiently, and we can use it for operations such as indexing or modifying a particular component as an entity.

Document systems: From post office to library to office

Though the systems described in this issue are at various stages of research and development, the fruits of these projects will increasingly affect our daily dealings with documents. With declining costs and typical recognition rates in excess of 99 percent, OCR systems will become common, simplifying the storage, search, and excerpting of paper-based documents. Page layout analysis techniques will recognize a particular form or magazine format and allow its duplication. Diagrams will be input from pictures or by hand and logically edited. Pen-based computers will translate handwritten entries into electronic documents. Archives of paper documents in libraries and engineering companies will be electronically converted for more efficient storage and instant delivery to a home or office computer.

Though these changes are inevitable, paper documents will be with us to some degree for many decades to come. The difference will be their eventual integration into our computerized world.

This issue contains articles covering a wide range of applications in document image analysis, including purely textual systems, graphical systems, and systems that deal with both. Some of the same techniques are used across applications, but where special or new techniques are matched to a particular application, the authors provide descriptions of greater depth. Whether the application is for the postal service, library, office, manufacturing plant, or home, this issue should make it clear that the need for document image analysis is widespread and touches us each day. ■

Acknowledgments

We thank the many reviewers who contributed to the selection of articles for this issue. On average we received 10 reviews per article for the 29 submissions, and most included detailed comments to improve quality, as well as suggestions for further research. We also thank Jon Butler for his aid in developing and structuring this special issue. Finally, to the *Computer* staff, to our secretaries Ann Shovlin and Elaine Smiles, and especially to Jon Butler's assistant Carla Shaw, we express special appreciation for efficiently dealing with the voluminous paperwork still associated with conventional document processing.

Additional information

Machine Vision and Applications will publish a special issue in August 1992, titled "Document Image Analysis Techniques." That issue will delve more deeply into some of the techniques used in document analysis, including some of the systems described in this issue. It also includes a comprehensive bibliography of publications in this field compiled by the coeditors.

The International Conference on Document Analysis and Recognition is devoted solely to this field. The 1991 conference was held in St. Malo, France, under sponsorship of the International Association for Pattern Recognition (IAPR) and the IEEE Computer Society. The next conference will be held in Japan in October 1993.

A previous IAPR International Workshop on Structural and Syntactic Pattern Recognition (SSPR) concentrated on document processing. It was held in Murray Hill, New Jersey, in June 1990. The next SSPR workshop will be held in Bern, Switzerland, in August 1992.

Document analysis research will also be found in proceedings from the International Conference on Pattern Recognition (ICPR) and the Frontiers of Handwriting workshop, both sponsored by the IAPR and IEEE Computer Society.

Lawrence O'Gorman has been with the Computing Systems Research Laboratory of AT&T Bell Laboratories, Murray Hill, New Jersey, since 1984 and is involved in the design of the RightPages electronic library project. From 1980 to 1981, he was with Computing Devices Company in Ottawa, Canada, where he worked on digital signal processing and filter design. His research interests include image processing, pattern recognition, document image analysis, and machine vision precision.

O'Gorman received a BASc degree from the University of Ottawa, Ontario, in 1978, an MS degree from the University of Washington, Seattle, in 1980, and a PhD degree from Carnegie Mellon University, Pittsburgh, in 1983, all in electrical engineering. He is a member of the IEEE Computer Society.

Rangachar Kasturi is an associate professor of electrical and computer engineering at Pennsylvania State University. During the 1989-1990 academic year, he was a visiting scholar at the Artificial Intelligence Laboratory of the University of Michigan. His research focuses on applications of image analysis and computer vision techniques.

Kasturi is coauthor of the tutorial text *Computer Vision: Principles and Applications*, published by the IEEE Computer Society Press in 1991. He is also a contributing author and coeditor of *Image Analysis Applications*, and an associate editor of *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Kasturi received a BE in electrical engineering from Bangalore University, India, in 1968 and an MSEE and a PhD from Texas Tech University, Lubbock, in 1980 and 1982, respectively. He is a senior member of the IEEE and a member of the Society of Photooptical Instrumentation Engineers, the ACM, and the IEEE Computer Society.

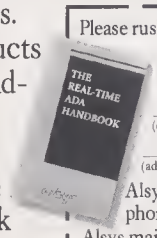
O'Gorman's address is AT&T Bell Laboratories, Room 3D-455, Murray Hill, NJ 07974, e-mail log@research.att.com. Kasturi can be reached at Pennsylvania State University, Electrical and Computer Engineering Dept., University Park, PA 16802. His e-mail address is kasturi@cmpe.psu.edu.

Now Available:
Alsys Ada Compilers
for the HP 9000 Series

If you don't think our real-time systems have taken off, consider some of the programs we've landed.

You already know the significance of the Alsys name when it comes to quality Ada. But you may not have realized that over half of our business is in—and over half of all our resources are devoted to—real-time applications. Right now Boeing, Lockheed, McDonnell Douglas, NASA, AIRBUS, European Space Agency and many others rely on us for solutions for mission and safety-critical applications. Alsys offers one of the broadest ranges of compiler products and developer tools, plus unparalleled expertise and guidance every step of the way.

If you need to reduce risk, deliver projects on time, and do both within budget, the first step is easy: send in the coupon, or call 617-270-0030. We'll rush you a handbook detailing some powerful reasons why you should consider enlisting our support. Alsys. We're serious about real-time.



Please rush me the Alsys handbook detailing your real-time capabilities.

(name) (title)

(company) (phone no.)

(address)

Alsys, Inc., 67 South Bedford Street, Burlington, MA 01803-5152,
phone: 617-270-0030 fax: 617-270-6882

Alsys maintains major operations in the U.S., Sweden, France, England, Germany and Japan.

alsys
A Thomson–CSF company

IE

Reader Service Number 1

A Prototype Document Image Analysis System for Technical Journals

George Nagy, Rensselaer Polytechnic Institute
Sharad Seth, University of Nebraska at Lincoln
Mahesh Viswanathan, IBM

Intelligent document segmentation can bring electronic browsing within the reach of most users. The authors show how this is achieved through document processing, analysis, and parsing the graphic sentence.

Let's quickly calculate the requirements of electronic data storage and access for a standard library of technical journals. A medium-sized research library subscribes to about 2,000 periodicals, each averaging about 500 pages per volume, for a total of one million pages per year. Although this article was output to film at 1,270 dpi (dots per inch) by an imagesetter, reproduction on a 300-dpi laser printer or display would be marginally acceptable to most readers (at least for the text and some of the art). At 300 dpi, each page contains about six million pixels (picture elements). At a conservative compression ratio of 10:1 (using existing facsimile methods), this yields 80 gigabytes per year for the entire collection of periodicals. While this volume is well beyond the storage capabilities of individual workstations, it is acceptable for a library file server. (Of course, unformatted text requires only about 6 kilobytes per page even without compression, but it is not an acceptable vehicle for technical material.)

A 10-page article can be transmitted over a high-speed network, and printed or displayed in *image form* in far less time than it takes to walk to a nearby library. Furthermore, while *Computer* may be available in most research libraries, you may have to wait several days for an interlibrary loan through facsimile or courier. There is, therefore, ample motivation to develop systems for the electronic distribution of digitized technical material.

However, even if the material is available in digital image form, not everyone has convenient access to a high-speed line, a laser printer, or a 2,100 × 3,000-dot display. We show how *intelligent document segmentation* can bring electronic browsing within the reach of readers equipped with only a modem and a personal computer.

Document analysis constitutes a domain of about the right degree of difficulty for current research on knowledge-based image-processing algorithms. This is important because document analysis itself is a transient application: There is no question that eventually information producers and consumers must be digitally linked. But there are also practical advantages to recognizing the location and extent of significant blocks of information on the page. This is also true for segmenting and

0018-9162/92/0700-0010$03.00 © 1992 IEEE

Glossary

AND-OR graph (or tree). Representation of a solution strategy in which a path from the start node to the solution node requires traversing any branch at an OR node and every branch at an AND node. In a related Min-Max search used in two-person games, a path from the start node to the solution node takes the lowest cost branch at a Min node and the highest cost branch at a Max node.

Bitmap. Digital representation of an image in which points are mapped to an array of binary pixels.

Branch-and-bound. A search technique that avoids paths certain to lead to higher cost solutions than the best solution obtained so far.

CCITT (Comite Consultatif International de Telegraphie et Telephonie). International organization which promulgates standards for facsimile coding and transmission. CCITT Group 3 compression methods are used over relatively high-error-rate channels such as the telephone network. Group 4 standards are designed for low-error-rate channels such as public data networks. The compression ratio of a code is the number of bits in the bitmapped representation of the image divided by the number of bits in the coded representation.

Compiler tools. Programs that generate lexical and syntactic analysis programs for particular applications, including assembly, interpretation, compilation, and translation of computer programs. In the Unix system, Lex is a popular tool for lexical analysis and YACC (yet another compiler compiler) is used for parsing context-free languages.

Document interchange formats. Standards for coding the organization of the layout and contents of a document to facilitate transferring documents from one computer system to another. Popular examples include the Document Content Architecture (DCA), Document Interchange Format (DIF), Document Style Semantics Specification Language (DSSSL), Office Document Architecture (ODA), and Standard Generalized Mark-up Language (SGML).

Drop cap. An oversized uppercase character spanning several printed lines, often used to begin an article or a section.

Hypertext. Data structure for text or other media that includes linkages (pointers) between conceptually related items. These linkages allow considerable latitude in selecting strategies for traversing many levels of information. The linkages are either preset or created by data users.

Kerning. Separate characters placed closer together than usual by removing some of the space between them.

Layout. Physical (for example, geometric and typographic) organization of textual and graphic matter in a document, in contrast to content.

Leading. The separation between consecutive lines of text, originally achieved in printing by inserting strips of lead between rows of type.

Lexical analysis. Each token (word) belongs to a lexical category that determines what tokens may precede or follow it. In English, examples of lexical categories are verb, noun, and pronoun. Lexical analysis determines the category of a token.

Ligature. Two or more symbols, such as æ, printed as a single pattern.

OCR (optical character recognition). The technology of converting printed or written material into computer-readable (ASCII) code. The word *optical* serves to distinguish it from magnetic ink character recognition (MICR). OCR systems usually include an optical scanner, a preprocessor to locate the characters in the appropriate order, a pattern classifier, and a contextual postprocessor.

Page icon. A reduced-scale representation of the subdivision of a printed page into nested rectangles that correspond to significant layout units. It is a graphic display of the X-Y tree of the decomposition.

Point. Printer's unit of measurement, equal to about 1/72 of an inch (US, European, and Belgian points are slightly different). One pica equals 12 points. The notation 10/12 indicates 10-point type with 2-point leading (white space).

PostScript. A programming language introduced by Adobe Systems that allows device-independent specification of typeset text and illustrations for computer printers, displays, and digital phototypesetters. Such languages are generally known as page-description languages.

Syntax. Synonymous with grammar: the formal rules that determine the permissible configuration of lexical tokens such as words. Parsing (a form of syntactic analysis) determines whether a string of words forms a legal sentence. The language accepted by a grammar is the (often infinite) set of all legal sentences. A grammar is usually expressed as a set of rewrite rules, or productions, that allow replacing the string of symbols found on the left-hand side of the production by the string found on the right-hand side to generate legal sentences. Symbols that don't occur by themselves on the left-hand side and therefore cannot be replaced are called terminal symbols; all others are called nonterminal symbols. Formally, a grammar is a quintuple consisting of a starting symbol, a delimiter, a set of terminal symbols, a set of nonterminal symbols, and a set of productions.

TIFF (tagged image-file format). A family of popular formats for color, gray-level, and black-and-white images in either compressed or uncompressed form.

VGA (video graphic adapter). A format convention for 480 × 640-pixel color screen displays introduced by IBM in 1987.

X-Y tree. A spatial data structure in which each node corresponds to a rectangular block. The children of each node represent the locations of the subdivisions of the parent block in a particular (horizontal or vertical) direction. The same organization is often represented in VLSI design by means of a polar graph. Other popular hierarchical data structures for isothetic (Manhattan) spatial subdivisions include the K-D tree and the Quad tree.

Advantages of intelligent page segmentation

Smaller blocks can be reproduced on PC displays without horizontal scrolling or loss of legibility.

Selected blocks can be transmitted more quickly than an entire page can for browsing over networks.

Layout analysis helps OCR preserve the reading order.

Key-entry of selected blocks that are intractable by OCR is less costly than rekeying entire pages.

Differentiating text from graphics and halftones leads to more efficient image compression.

Straight borders between high-contrast and halftone regions are preserved for digital reprographics.

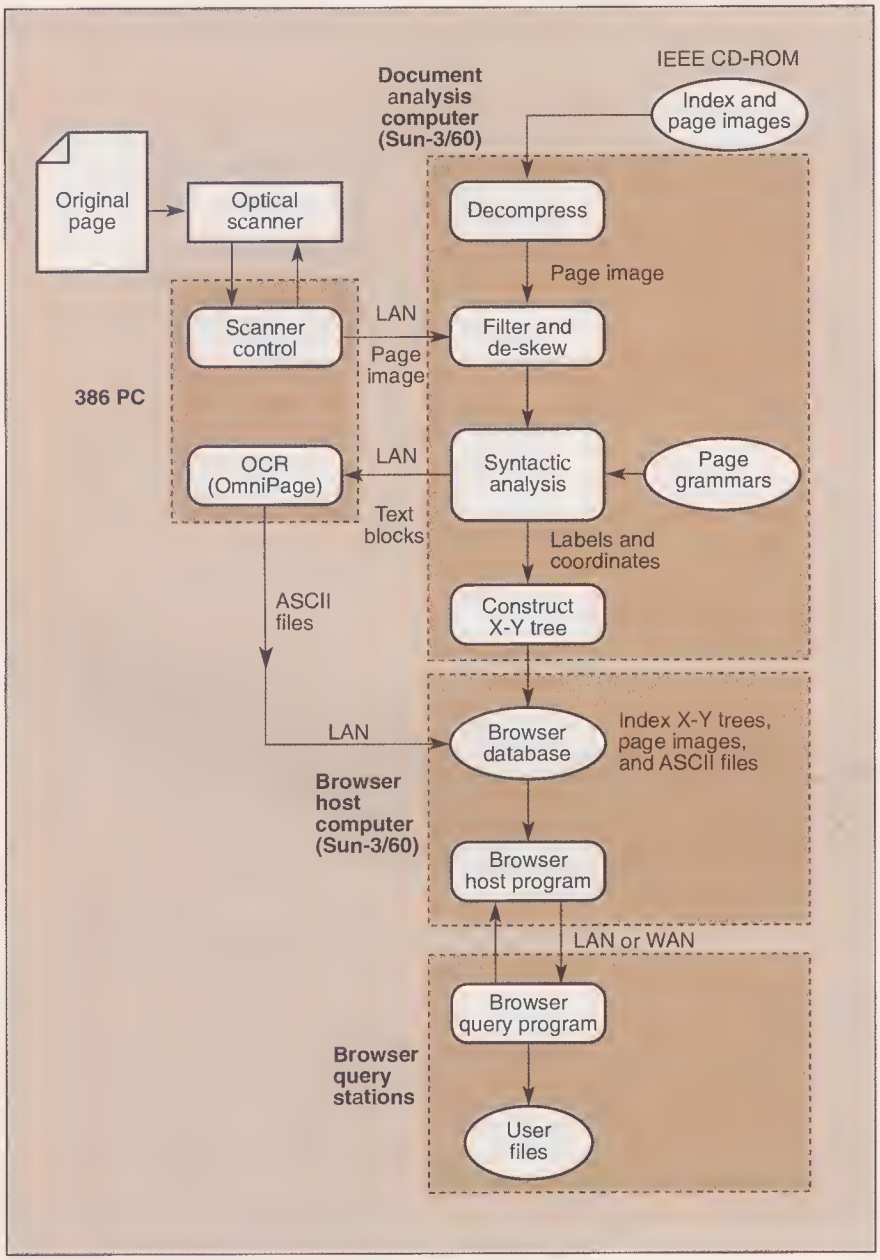


Figure 1. Gobbledoc system overview.

labeling the page image according to its logical structure, without resorting to optical character recognition (see the "Advantages of intelligent page segmentation" sidebar).

Research on OCR began at the turn of the century. Inventions for replacing telegraph operators and assisting blind readers were first demonstrated in 1914, and high-speed OCR systems have been commercially available since 1955 for specially printed forms and fixed-pitch typescript. The field of digitized page analysis started in the sixties, but only recently have the relevant technologies matured sufficiently to allow the conversion of complex typeset documents such as technical articles. Major catalysts include accurate, fast, inexpensive page scanners; high-speed computer networks; sufficient storage and processing power for digitized page images; and compression standards for digital facsimile.

Document image analysis requires the assembly of a number of software tools.[1] Gobbledoc, the system that we have developed at Rensselaer Polytechnic Institute over the last five years, integrates several existing components with a new method of image decomposition. Figure 1 shows the flow of information from the original document to the user's computer screen. A page is either scanned locally or obtained from a CD-ROM. It is segmented and labeled by using a syntactic approach, and stored in an X-Y tree. Images of text blocks are transferred to a personal computer and recognized by using OCR. The resulting ASCII output (which includes some special characters marked with escape quotes) is linked to the corresponding node of the X-Y tree. All data is then stored on the browser host computer. The browser is activated from a remote client workstation, and the user-requested information is displayed on the client terminal.

In Gobbledoc, image processing, document analysis, and OCR operations take place in batch mode when the documents are acquired. After explaining the document image acquisition process, we describe the knowledge base that must be entered into the system to process a family of page images. We show how our X-Y tree data structure converts the two-dimensional page-segmentation problem into a series of one-dimensional string-parsing problems that can be tackled by using con-

ventional compiler tools. Next, syntactic analysis is used to divide each page into labeled rectangular blocks. Blocks labeled *text* are converted by OCR to obtain a secondary (ASCII) document representation. But such symbolic files are better suited for computerized search than for human access to the document content. Because too many visual layout clues are lost in the OCR process (including some special characters), our system preserves the original block images for human browsing. Finally, we consider storage, networking, and display issues specific to document images, and describe our prototype browser.

Partial layout specifications for *Computer* in 1991

The title lines are set in Melior 36/38 point boldface, centered.

There are one to four lines in the title: the second part of a long title is sometimes set in 24-point type.

The byline is set in Melior 12/14-point boldface, centered; affiliations may either follow the authors' names set in the same typeface or are set on the same line.

The title line precedes the byline, and the two are separated by at least 38-point leading.

Every paragraph is indented except the first and the beginning of the conclusion section, which begin with a 40/40-point drop cap.

The body type (main text) is TimesTen Roman 9/11.

The page numbers and month-year (in body type) are set flush with the margin and alternate between left and right.

Footnotes (rare) are set TimesTen Roman 7/8, separated by a hairline rule with a blank line before and after the rule.

Preprocessing

We convert each page to digital form by scanning it at 300 dpi. This rate is sufficient to preserve all significant white spaces. Since the entire X-Y tree approach is extremely sensitive to skew, we carefully align each page on the scanner bed. We recognize that this could be impossible in a production environment, where the systems could incorporate one of the excellent skew-correction methods (performed by software or hardware) already demonstrated by others.[2]

Sample pages. We also take sample pages from the CD-ROM database prepared for the IEE-IEEE by University Microfilms. The IEEE annually distributes about 130,000 pages in 300-dpi image form on CD. Other professional societies are not far behind. The IEE-IEEE database is stored in CCITT Group 4 compressed form. This is the two-dimensional Modified Modified Read (MMR) code developed as a standard for facsimile terminals operating over very-low-error-rate public data networks.[3] Currently, we decompress the page using software, but standard chips for this purpose are available. Since we have no control over the degree of skew of the CD pages, we de-skew them.

Noise. Our documents contain specks of noise caused by flawed fibers in the stock, imperfect printing, photocopying, and digitization. (High-quality journals are relatively noise-free, but we used photocopies on uncoated paper.) Such noise does not bother human readers, but it complicates automated analysis. To cope with the noise, one may construct page grammars sufficiently robust to ignore speckle. This is feasible but tedious. Instead, we filter out all specks smaller than a given size in a preliminary pass. After the analysis is completed, all specks are restored. The size threshold therefore can be quite generous. Losing a few periods or dots on the *i*'s and *j*'s does not affect the layout analysis, and they are restituted before any document component is submitted to OCR or displayed for human inspection.

Layout encoding

The first step in analyzing a specific family of documents is encoding the information that distinguishes this family from others. The required knowledge base is quite specific. It may be stated explicitly in the form of if-then rules, a form-definition language, a geometry tree of box locations, an expert system, or a formal grammar — or implicitly embedded in the processing programs. Some examples of constraints for the title page of feature articles in *Computer* as of July 1991 are shown in the accompanying sidebar. The rules for the pages from *IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE-PAMI)*, used in our experiments and shown later in Figure 6, are similar.

Publication-specific systems have been successfully demonstrated on newspaper pages, business letters, articles, and tables of contents in technical journals, patent applications, resumes, typed forms with a prespecified layout, sheet music, maps, engineering drawings, circuit diagrams, and even the periodical *Chess Informant*. A complex knowledge-based system that demonstrates the usefulness of full interaction between different components has been developed for postal address location and interpretation. Recent achievements in these areas are described in the proceedings of conferences on document analysis.[4,5]

Syntactic analysis. Just as every programming language has its fixed set of rules, each publication has predetermined layout conventions. These conventions dictate the size, position, spacing, and ordering of the blocks that correspond to logical entities on a page, as shown in the sidebar for *Computer*. Accordingly, documents that satisfy the postulated layout conventions can always be parsed into different components without OCR in the same way that syntactically correct program code can be parsed into meaningful constructs without semantic understanding.

Syntactic analysis accomplishes both segmentation and component identification simultaneously. A publication-specific *document grammar* is a formal description of all legal page formats that articles in a given publication can assume. In principle, a parse of the page will reveal whether the page is acceptable under the rules of the grammar and, if so, what labels must be assigned

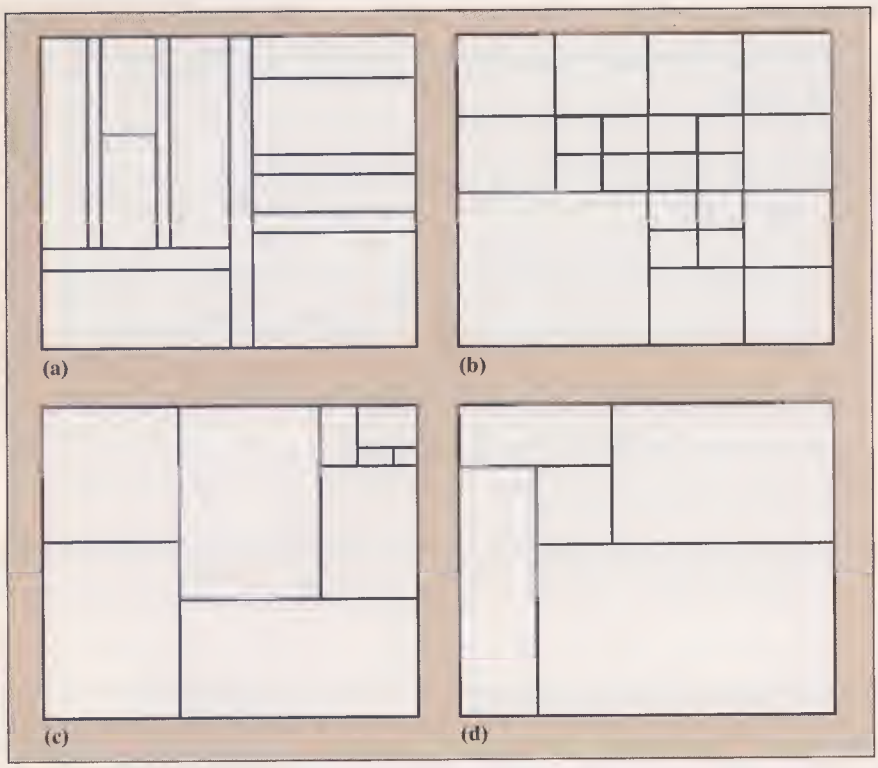


Figure 2. Hierarchical subdivision of rectangles into rectangles: X-Y tree for page segmentation (a); Quad tree for comparing or combining several images (b); K-D tree for fast search (c); example of tiling with rectangular blocks that cannot be obtained by successive horizontal and vertical subdivisions only (d).

to the various components to construe a valid graphic "sentence."

Although we use compiler tools developed primarily for formal languages, the syntactic analysis of document images exhibits many of the difficulties of parsing natural language. Layout conventions may be insufficient to identify *every* document component. For instance, text lines with equations buried in them may radically alter the expected line spacing. We must therefore ensure that minor deviations have only local effects. Furthermore, the grammar for a modern programming language is established from the start, while document grammars must be inferred indirectly, as later discussed. (A never-ending task: Journals frequently put on new faces.)

Block grammars. The document grammar for a specific journal consists of a set of block grammars. Each block grammar subdivides a block horizontally or vertically into a set of subblocks. The net result of applying the entire document grammar is therefore a subdivision of the page into *nested rectangular blocks*. Such a subdivision can be represented efficiently in a data structure called the X-Y tree[6] (Figure 2). The block grammars themselves are also organized in the form of a tree: The block grammar to be used to subdivide each block is determined recursively by the results of the parse at the level above.

Syntactic attributes

A (horizontal/vertical) *block profile* is a binary string that contains a zero for each horizontal or vertical scanline that contains only white pixels; otherwise it is a one.

A *black atom* is a maximal all-one substring. It is the smallest indivisible partition of the current block profile. A *white atom* is an all-zero substring.

A *black molecule* is a sequence of black and white atoms followed by a black atom. A *white molecule* is a white atom that separates two black molecules.

An *entity* is a molecule that has been assigned a *class label* (title, authors, figure caption). It may depend on an ordering relationship.

This approach effectively transforms the difficult two-dimensional segmentation into a set of manageable one-dimensional segmentation problems.

The syntactic formalism is theoretically well understood, and sophisticated software is available for lexical analysis and parsing of strings of symbols. Each block grammar is therefore implemented as a conventional string grammar that operates on a binary string called a *block profile*. The block profile is the thresholded vertical or horizontal projection of the black areas within the block. Zeros in the block profile correspond to white spaces that extend all the way across the block and are therefore good candidates for the locations of subdivisions.

Representing the structure of an entire page in terms of block grammars simplifies matters considerably. But each block grammar itself is a complex structure. It must accommodate many alternative configurations. For instance, to divide the title block from the byline block, the block grammar must provide for a varying number of title lines and bylines, and for changes in spacing caused by the ascenders and descenders of the letters. To simplify the design process, each block grammar is constructed in several stages, in terms of syntactic attributes extracted from profile features.[7]

Syntactic attributes. The first stage of a block grammar operates on the ones and zeros of the block profile. Strings of ones or zeros are called *atoms*. Atoms are divided into classes according to their length. A string of alternating black and white atoms is a *molecule*. The class of a molecule depends on the number and kind of atoms it contains. Finally, molecules are transformed into *entities* depending on the order of their appearance. The words *atom*, *molecule*, and *entity* were chosen because they are not specific to a particular publication or subdivision. (See the sidebar on syntactic attributes.)

The syntactic attributes that determine the parse are the size and number of atoms within an entity, and the number and order of permissible occurrences of entities on a page. Table 1 shows the expected variation in the horizontal profile of a page fragment that includes the title and byline. The assignment of symbols into larger units is accomplished by rewriting rules or *productions*. These

Table 1. Examples of syntactic profile features for *Computer* digitized at 300 dpi.

	Title	Title-byline-space	Byline
Black atom length	100-160	—	35-52
White atom length	4-20	10-40	2-14
No. atoms	1-4	—	1-6
No. entities	1	1	1
Precedence		Title before byline	

rules take into account the expected variability. The entries in the first row correspond to the maximum and minimum height (in pixels) of the title and byline lines in a feature article scanned at 300 dpi. The second row shows the expected profile height of the leading. There must be one to four title lines and one to six author lines, with the byline below the title. Each page has only a single title block and a single byline block.

In the greatly simplified example of Figure 3, a typical horizontal block profile is shown at the top. The atom lengths have been shortened to show the entire horizontal block profile.

In stage 1, runs of ones or zeros are condensed into atoms according to their length. For example, black runs of 2 to 3 are called a. In stage 2, atoms are grouped into molecules. (A run of zero to three black-white sequences of two atoms of types b,v or b,w, followed by a black atom of type b, is labelled A.) In stage 3, molecules are interpreted as *document entities.*

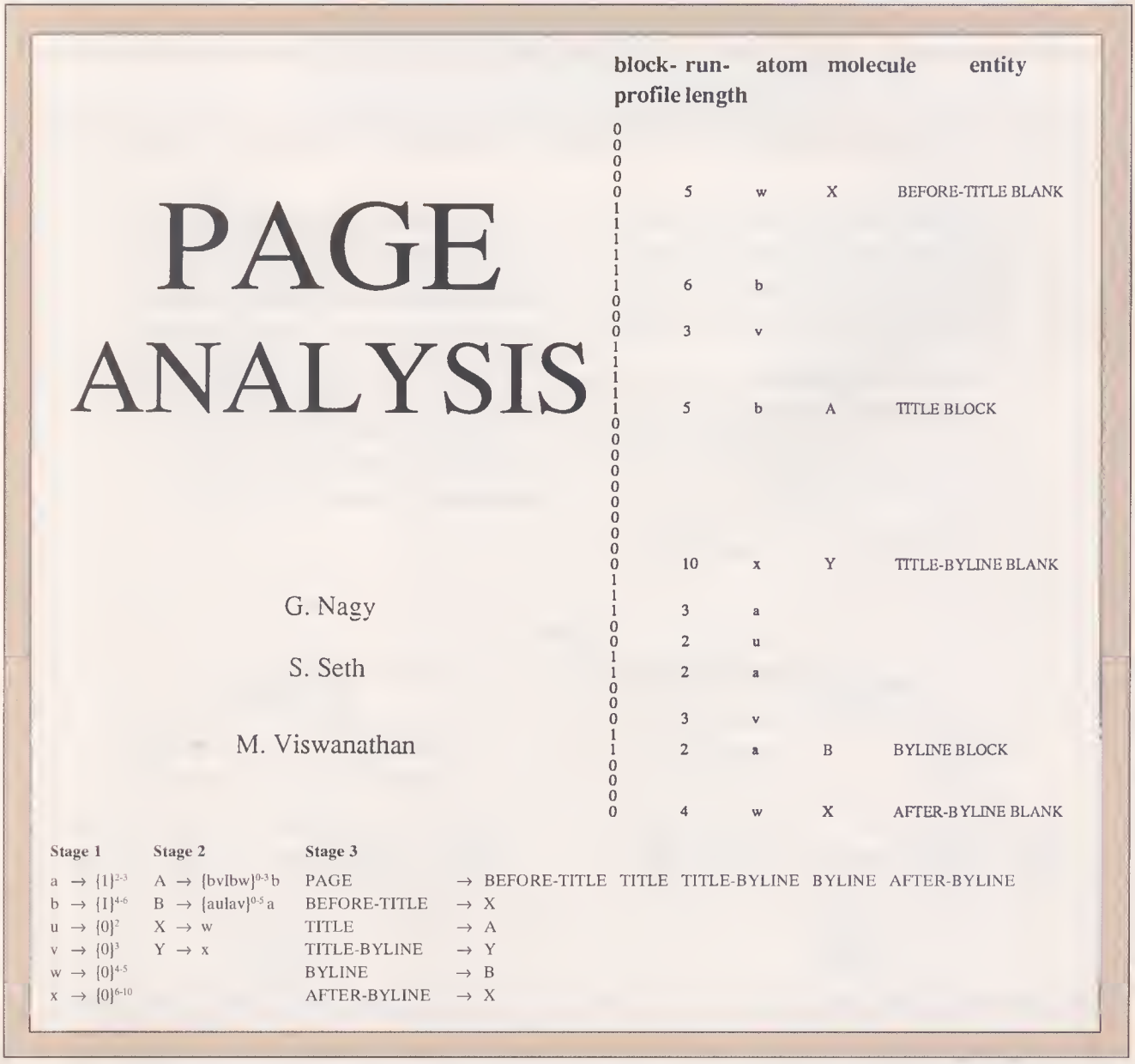


Figure 3. Illustration of a simplified block grammar for the first vertical subdivision of a title page similar to that of *Computer*.

The block grammar at the bottom specifies that the title block (molecule A) may have one to four lines of print, separated by blank lines. Each line of print (atom b) may be four to six pixels high, and each blank line may be three (v), four, or five (w) pixels high. (The height of the blank lines is divided into two ranges because of the overlap in the range of the heights of the blank lines in the title and author blocks.) Stage 3 of this block grammar ensures that the title block precedes the byline block.

Note that molecule X is sometimes interpreted as a before-title blank and sometimes as an after-byline blank, depending on its location with respect to other molecules. Stage 4 (not illustrated) merges consecutive entities with the same label separated by a wide blank, such as text paragraphs.

Block grammars based on such attributes typically have hundreds of productions for a single publication such as *IEEE-PAMI*. Defining each block grammar directly on the binary profile strings, while theoretically equivalent, would be too cumbersome in practice.

Grammar specification. Constructing the block grammars by hand is still very time-consuming and is the greatest weakness of our current approach. Available sources of information are computer-aided measurements on samples of scanned copy, publishers' style manuals, and macro definitions customized for a given publication in page formatters. We have also developed a tabular representation of layout parameters that is automatically translated into block grammars.

Appealing alternatives to our tabular representation of document layout include the Office Document Architecture (ODA), Standard Generalized Mark-up Language (SGML), and Document Style Semantics Specification Language (DSSSL) document interchange standards.[8] However, it is not easy to extract from these specifications the criteria for alternating horizontal and vertical subdivisions required by our decomposition.

Low-level block grammars tend to be much more generic (exhibit less publication-dependent variability) than high-level grammars. One can therefore often reuse existing block grammars for a new family of publications. There are, for instance, only a few paragraph formats in common use. Ideally, the initial parameters would be adjusted (learned)

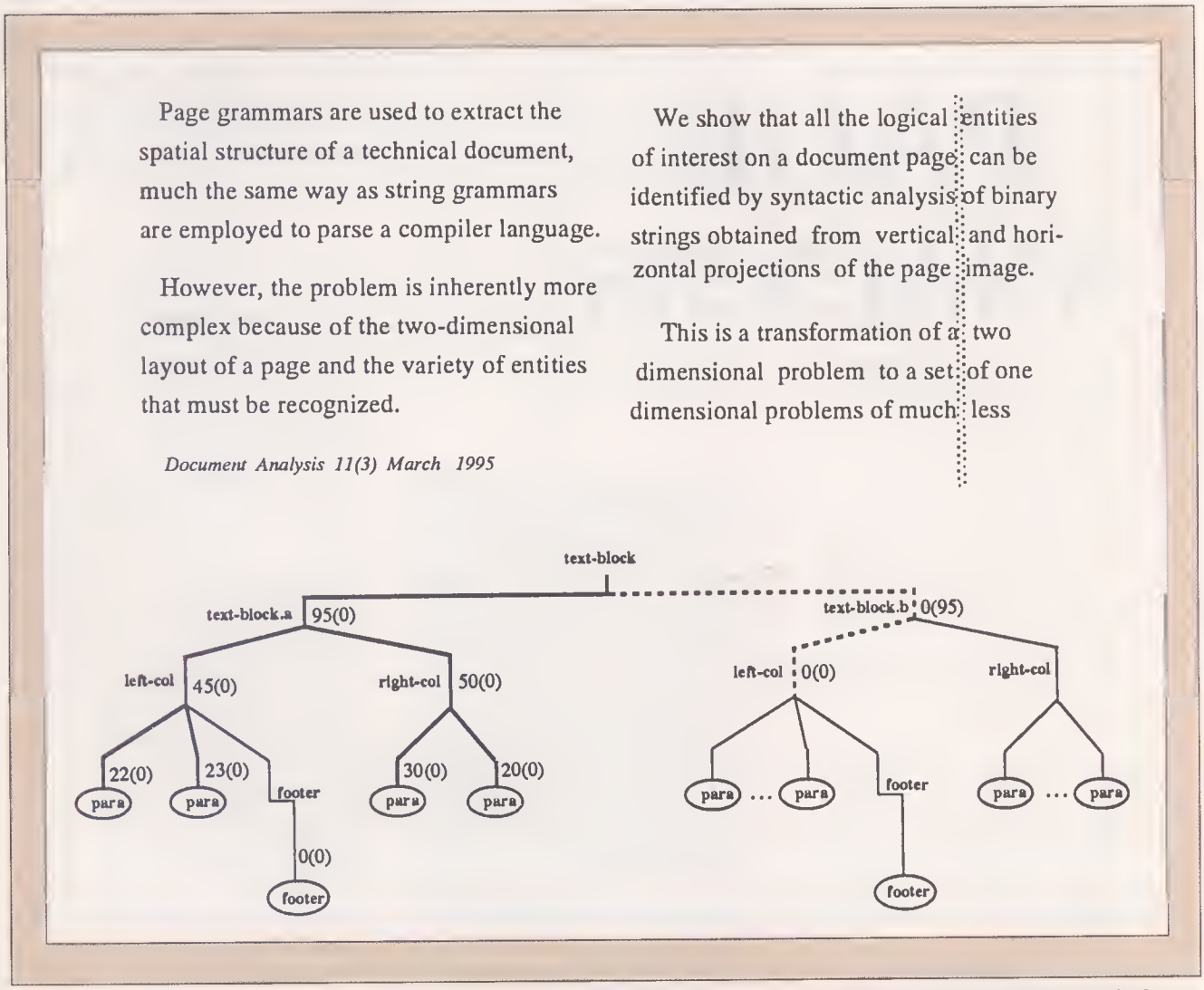


Figure 4. Simplified example of branch-and-bound analysis: a two-column text block with a *river* (accidental vertical alignment of blank spaces) (a), and a block grammar *text-block.a* dividing the block at the central gutter (b).

during actual operations; this is a subject of continuing research.

Page decomposition. We now show how the block-grammar concept can be extended to segment and label a whole page. A block grammar is intended to interpret each subblock of a given block as a particular (labeled) entity. At the subblock level, block grammars assign labels to abstract, title-block, byline-block, reference-entry, and figure-caption entities. Each of these lower level entities may have alternative grammars corresponding to different formats. During the parsing of a block, a unique label is associated with each subblock. Only blocks that are assigned labels for which no block grammars are provided are assumed to be correctly segmented and labeled. Every other *nonterminal* symbol assigned by a block grammar is equated with the start symbol of a block grammar at the level below. Unlike the four stages of each block grammar, the set of block grammars cannot be combined into a single master grammar, even in principle, because at each level a new set of block profiles must be extracted, and the location of each block boundary depends on the parse at the level above.

Several dozen different classes of logical components are isolated and identified by parsing the recursively generated block profiles. The recursive algorithm that subdivides the page verifies the correctness of label assignments in a depth-first traversal. It returns with success only if parsing is successful for the given block and all of its nested subblocks. The algorithm may try each of several alternative block grammars in turn. For example, if it is not known what part of an article a given page represents, it may be parsed with two block grammars: one designed for a title page and one for a nontitle page. The parse may therefore be considered an AND-OR tree: At each level, *every* subblock must be identified by means of one of a set of alternative grammars.

Another reason for alternative grammars is that a block profile may have two legal parses. This happens when the label of a block cannot be determined from its own profile, but only at a level below. In that case, the block must be parsed under several hypotheses, using alternative grammars.

Error control. Some portion of the page may be in an unexpected format for which no alternative grammar has been provided. Since this will yield an unrecognizable subblock at some point, the entire parse will fail. The AND-OR approach is therefore modified to avoid catastrophic failure and determine efficiently the *best possible labeling*. "Best" is defined as the *maximum cumulative area* of the labeled blocks. A *branch-and-bound* strategy avoids parsing a subblock if the maximum labeled area of the page cannot be increased over the current lower bound even with complete segmentation and labeling of the subblock (Figure 4). Subblock grammar left-col correctly divides the left column into two paragraphs and a footer, but the footer grammar fails because it expects even smaller type. Grammar right-col correctly processes the right column, so 95 percent of the area (all but the footer block) is now labeled.

In an attempt to improve on this, the second alternative top-level grammar, text-block.b (designed for narrower gutters), is applied to the entire block and divides it at the river. Now, however, left-col fails, because the lines of text do not line up across the central gutter, so the block cannot be segmented into horizontal strips at the prescribed line-spacing. The search is abandoned without attempting the rightmost branches of the tree, because even if the region to the right of the river were correctly labeled, the overall result would be inferior to the earlier parse. The numbers on the branches of the search tree show the percentage of area identified and, in parentheses, the applicable lower bound.

Typographic conventions. Our method uses a publication-specific knowledge base in the form of a document grammar. We have also studied segmentation techniques based on typesetting conventions[9] rather than on publication-specific knowledge. Previous methods generally imbed the typographic constraints in the processing routines. It is therefore difficult to take an existing program and form a clear conception of its capabilities without extensive experimentation. Further progress may depend on the development of more consistent and comprehensive knowledge bases through explicit codification of such information. Some examples of generic typesetting knowledge for technical journals set in derivatives of the Latin alphabet are shown in the sidebar called "Generic typesetting rules." The commercial software that we use for OCR incorporates such typesetting rules to locate and isolate characters in a block of text.

Generic typesetting rules

Printed lines are roughly horizontal.

The baselines of characters are aligned.

Each line of text is set in a single point size.

Ascenders, descenders, and capitals have consistent heights. In roman fonts, serifs are aligned.

Typefaces (including variants such as italic or bold) do not change within a word.

Within a line of text, word and character spaces are uniform, and word spaces are larger than character spaces.

Lines of text in a paragraph are spaced uniformly.

Each paragraph is left-justified or right-justified (or both), with special provisions for the first and last line of a paragraph.

Paragraphs are separated either by wider spaces than lines within a paragraph or by indentation.

Illustrations are confined to rectangular frames.

In multicolumn formats, the columns are spaced uniformly.

OCR

Rather than develop our own OCR, we considered commercially available products.[10-12] High-end systems consist of a combination of hardware and software. Some require only an additional processor board. Low-end systems are software only and typically run on PCs with extended internal storage. Some OCR systems can be trained for specific typefaces.

We selected the OmniPage system from Caere Corp. because of its superior performance (without user training) on typefaces and sizes used in technical journals. Nevertheless, the ASCII version produced by the OCR system is not as accurate as a keyed-in version. Unusual typefaces may baffle the system, but most of the classification errors are due to missegmented characters (particularly in small point sizes, and kerned, bold, underlined, and italicized typefaces). Formulas and equations are sometimes mangled. Lines with drop caps (such as the first letter of this article), large mathematical symbols, or superscripts may be missed altogether. Some mistakes seem irrational and may be repeated many times.

In other applications, where accuracy is essential, the text would be corrected using a spelling checker and human postediting. However, even without expensive postprocessing, the ASCII version obtained by OCR is useful not only for automated search but also because it can be converted into a standard document-interchange format[8] or copied directly into a user's word processor or desktop publishing system. (We leave aside the difficult questions of pricing, copyright law, and plagiarism.)

Text-image compression. We are also attempting to use the output of the OCR program to provide a highly compressed version of the text image. Only the first graphic instance of each character pattern is stored. When a subsequent pattern is identified by the OCR with the same alphabetic label, the bitmaps of the original pattern and the new pattern are compared, and if they match, only a pointer to the original bitmap is stored. The degree of match required governs the fidelity of reproduction. Earlier symbol-matching schemes carried out bitmap comparisons with all previous prototype patterns. We use, instead, an efficient OCR system for pattern identification. On dense all-text printed pages scanned at 300 dpi, preliminary results indicate compression ratios twice as high as CCITT Group 4.

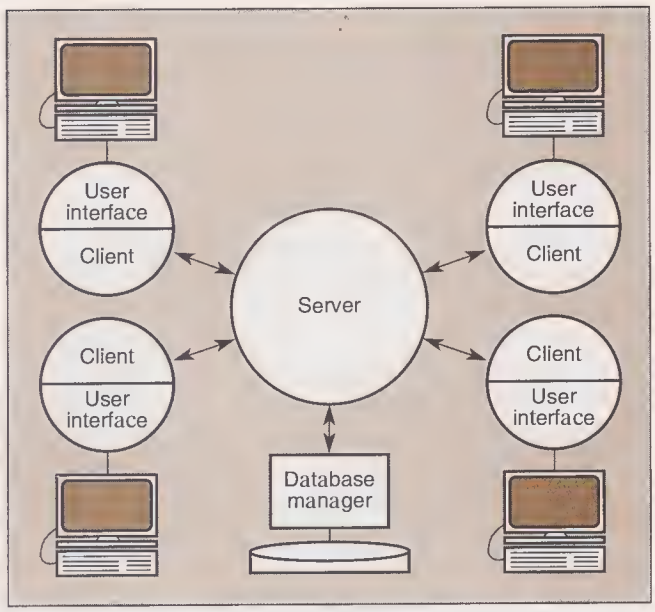


Figure 5. The remote browsing system consists of a central server connected to several query stations by either a local or a wide area network.

Document browser

Remote image browsing systems often use some form of progressive encoding and transmission. Initially, only a low-resolution image is displayed. Greater detail then emerges as additional data is transmitted. Such a scheme is almost useless for printed documents, since a low-resolution version is nearly illegible. Instead, we display parts of the page on demand, using the previously obtained subdivision into meaningful blocks.

The browser system consists of two programs on separate computers that communicate. One is a host program that handles requests from remote users and manages the database of digitized documents, labeled X-Y trees, and ASCII renditions of the textual leaf nodes of the X-Y trees. The other is a (query-station) client program that generates the user requests. The portions of the labeled image requested by the user display at the client station. For portability, we have also implemented an X-Windows version of the browser (see Figure 5).

The host program consists of the server and a database manager. The server handles the communication between the host and the query station. The query station consists of a client and a user interface. The interface transfers a user's request to the client. The client establishes a connection to the host and submits the request. The database manager parses the request received by the server at a specified port and sends the necessary data to the query station. The data received from the server is then transmitted to the user.

The user selects the desired page from a pop-up menu. A geometric representation of the page (a scaled-down labeled X-Y tree with only leaf nodes) displays. The desired block is then retrieved by "mousing" the block in the geometric representation. Using the ASCII button, the user can display either an ASCII version of the text block or a 300-dpi image.

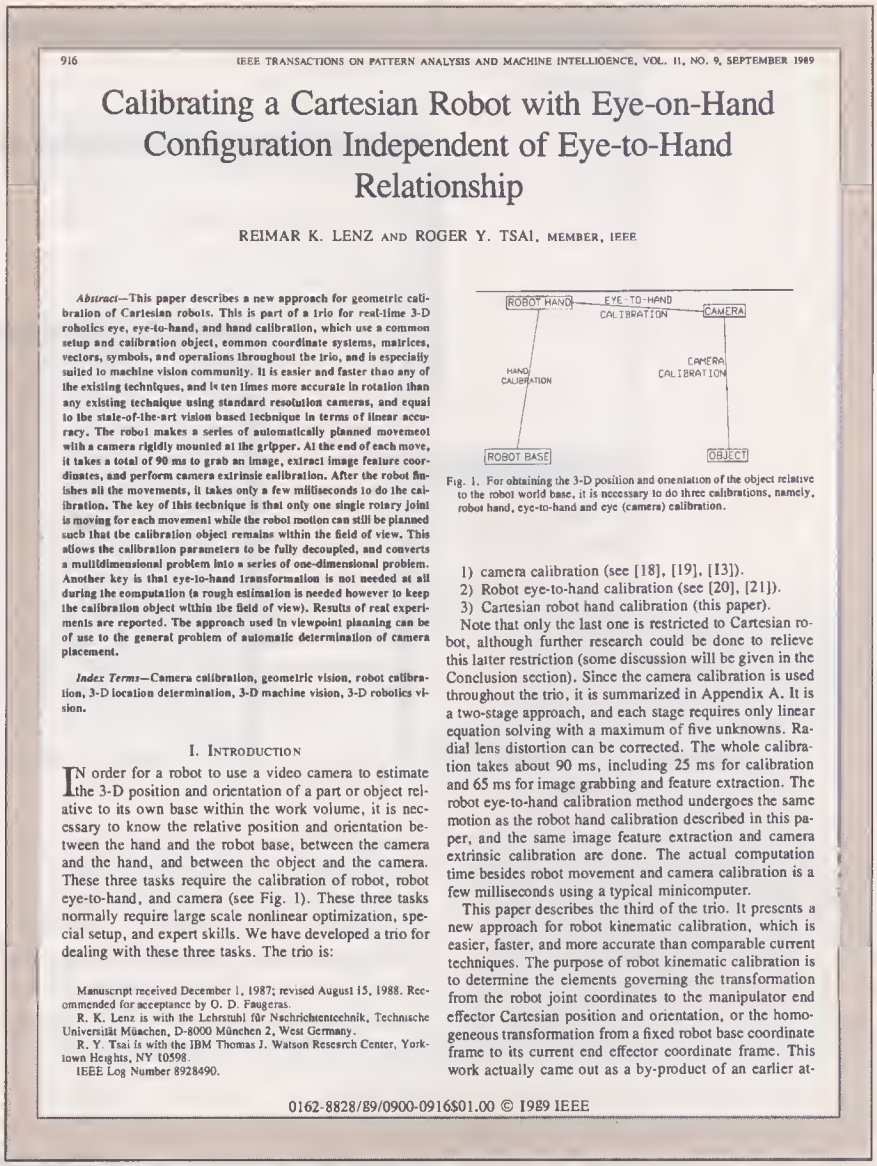
Line wrapping. Most technical printed material is laid out in columns narrow enough for display at 300 dpi on a 480 × 640-pixel VGA screen. However, some type — often the title and abstract — spreads across the full page and cannot be displayed in image form without unwieldy horizontal scrolling. When a text block is wider than a preset width, we continue the segmentation process to the line level. We then locate interword blanks which, in each line of text, are wider than the widest intercharacter space. Each text line is divided at one or more word boundaries and consecutive segments are displayed under one another. This is a form of line-wrapping for text in *image*, rather than symbolic, form.

Linkages. The dual text-image representation provides opportunities to enhance document access. For instance, the OCR system can identify all instances of the word "Figure" in the text and in figure captions. Since we know the location of each figure and figure caption, we can derive the figure number of each illustration. Then, when the user encounters a mention of a figure in the text, depressing the mouse button brings the figure to the screen in a separate window. Alternatively, all figures mentioned in any text paragraph on the screen can be simultaneously displayed with reduced resolution.

Similar concepts apply to cited references. Again, the ASCII version is necessary to set the linkages automatically, but the references can be invoked from either image or ASCII text display. While we have experimented only with linkages within the same document, the concepts can be extended to multiple documents. Image processing plays a role here only in the identification of the coordinates of a recognized keyword in the image and in segmenting the page into blocks. Further enhancement of the information falls in the realm of hypertext.

Implementation. The decompression, de-skewing, noise removal, and syntactic analysis programs run on a Sun-3/60. The uncompressed bitmap of the entire page is stored only once. The profile extraction routines access the required blocks through the *x* and *y* coordinates stored after each parse in the X-Y tree. The parsers at each stage are C-language programs generated by the Unix compiler utilities Lex and YACC. A Unix shell program controls the recursion between levels. Generation and storage of the ASCII files is an off-line process. The MicroTek scanning program EyeStar, the OmniPage OCR system, and the CD-ROM access programs run on an Intel 80386 PC with 4 megabytes of memory.

The host and client programs of the remote browser also run on Sun workstations. All browser interactions are on-line processes. The prototype system has been tested with the host located at the Rensselaer Polytechnic Institute and the query station at the University of Nebraska, and vice versa. A single host can serve several query stations simultaneously. However, our browser does not have the elaborate information retrieval and bibliographic navigation facilities necessary for an operational system. At present, our browser contains only a few dozen pages, which can be selected from a simple menu. Furthermore, its response time is far too slow for anyone browsing in earnest. It does, however, demonstrate several functions related to document image analysis that would be desirable for full access to a technical library through a workstation.



916 IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 11, NO. 9, SEPTEMBER 1989

Calibrating a Cartesian Robot with Eye-on-Hand Configuration Independent of Eye-to-Hand Relationship

REIMAR K. LENZ AND ROGER Y. TSAI, MEMBER, IEEE

Abstract—This paper describes a new approach for geometric calibration of Cartesian robots. This is part of a trio for real-time 3-D robotics eye, eye-to-hand, and hand calibration, which use a common setup and calibration object, common coordinate systems, matrices, vectors, symbols, and operations throughout the trio, and is especially suited to machine vision community. It is easier and faster than any of the existing techniques, and is ten times more accurate in rotation than any existing technique using standard resolution cameras, and equal to the state-of-the-art vision based technique in terms of linear accuracy. The robot makes a series of automatically planned movemeot with a camera rigidly mounted at the gripper. At the end of each move, it takes a total of 90 ms to grab an image, extract image feature coordinates, and perform camera extrinsic calibration. After the robot finishes all the movements, it takes only a few milliseconds to do the calibration. The key of this technique is that only one single rotary joint is moving for each movement while the robot motion can still be planned such that the calibration object remains within the field of view. This allows the calibration parameters to be fully decoupled, and converts a multidimensional problem into a series of one-dimensional problem. Another key is that eye-to-hand transformation is not needed at all during the computation (a rough estimation is needed however to keep the calibration object within the field of view). Results of real experiments are reported. The approach used in viewpoint planning can be of use to the general problem of automatic determination of camera placement.

Index Terms—Camera calibration, geometric vision, robot calibration, 3-D location determination, 3-D machine vision, 3-D robotics vision.

I. Introduction

In order for a robot to use a video camera to estimate the 3-D position and orientation of a part or object relative to its own base within the work volume, it is necessary to know the relative position and orientation between the hand and the robot base, between the camera and the hand, and between the object and the camera. These three tasks require the calibration of robot, robot eye-to-hand, and camera (see Fig. 1). These three tasks normally require large scale nonlinear optimization, special setup, and expert skills. We have developed a trio for dealing with these three tasks. The trio is:

Manuscript received December 1, 1987; revised August 15, 1988. Recommended for acceptance by O. D. Faugeras.

R. K. Lenz is with the Lehrstuhl für Nachrichtentechnik, Technische Universität München, D-8000 München 2, West Germany.

R. Y. Tsai is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 8928490.

Fig. 1. For obtaining the 3-D position and orientation of the object relative to the robot world base, it is necessary to do three calibrations, namely, robot hand, eye-to-hand and eye (camera) calibration.

1) camera calibration (see [18], [19], [13]).
2) Robot eye-to-hand calibration (see [20], [21]).
3) Cartesian robot hand calibration (this paper).

Note that only the last one is restricted to Cartesian robot, although further research could be done to relieve this latter restriction (some discussion will be given in the Conclusion section). Since the camera calibration is used throughout the trio, it is summarized in Appendix A. It is a two-stage approach, and each stage requires only linear equation solving with a maximum of five unknowns. Radial lens distortion can be corrected. The whole calibration takes about 90 ms, including 25 ms for calibration and 65 ms for image grabbing and feature extraction. The robot eye-to-hand calibration method undergoes the same motion as the robot hand calibration described in this paper, and the same image feature extraction and camera extrinsic calibration are done. The actual computation time besides robot movement and camera calibration is a few milliseconds using a typical minicomputer.

This paper describes the third of the trio. It presents a new approach for robot kinematic calibration, which is easier, faster, and more accurate than comparable current techniques. The purpose of robot kinematic calibration is to determine the elements governing the transformation from the robot joint coordinates to the manipulator end effector Cartesian position and orientation, or the homogeneous transformation from a fixed robot base coordinate frame to its current end effector coordinate frame. This work actually came out as a by-product of an earlier at-

0162-8828/89/0900-0916$01.00 © 1989 IEEE

Figure 6. ***IEEE-PAMI*** **title page extracted from the IEEE CD-ROM database. A de-skewed version is used for OCR, and the image is also filtered for syntactic analysis.**

Example. A page from the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, extracted from the CD-ROM and printed on an AppleWriter from its PostScript representation, is reproduced in Figure 6. The image is de-skewed and filtered before generating a labeled X-Y tree for the page. The X-Y tree contains the coordinates of all segments in the page along with the assigned labels. The 12 subimages in the example that correspond to text are converted into a TIFF (tagged image-file format) and transferred to the PC using a network file-transfer protocol. In the example, these include page number, header, title, byline, abstract, keywords, section titles, two paragraphs of text, figure caption, footnote, and footer. OmniPage is run separately on each subimage and the corresponding ASCII output files are returned to the Sun system.

When the browser is activated, it gives the user successive choices of publication name, volume, issue, and page number. The browser control buttons are shown at the top left corner of Figure 7 on the next page. The final selection prompts the display of a labeled geometric representation of the X-Y tree of the selected page, as shown in the bottom left corner of the screen. Now, the user can choose between ASCII and image renditions of the processed blocks. Multiple requests are entertained, as is evident from Figure 7, which shows the

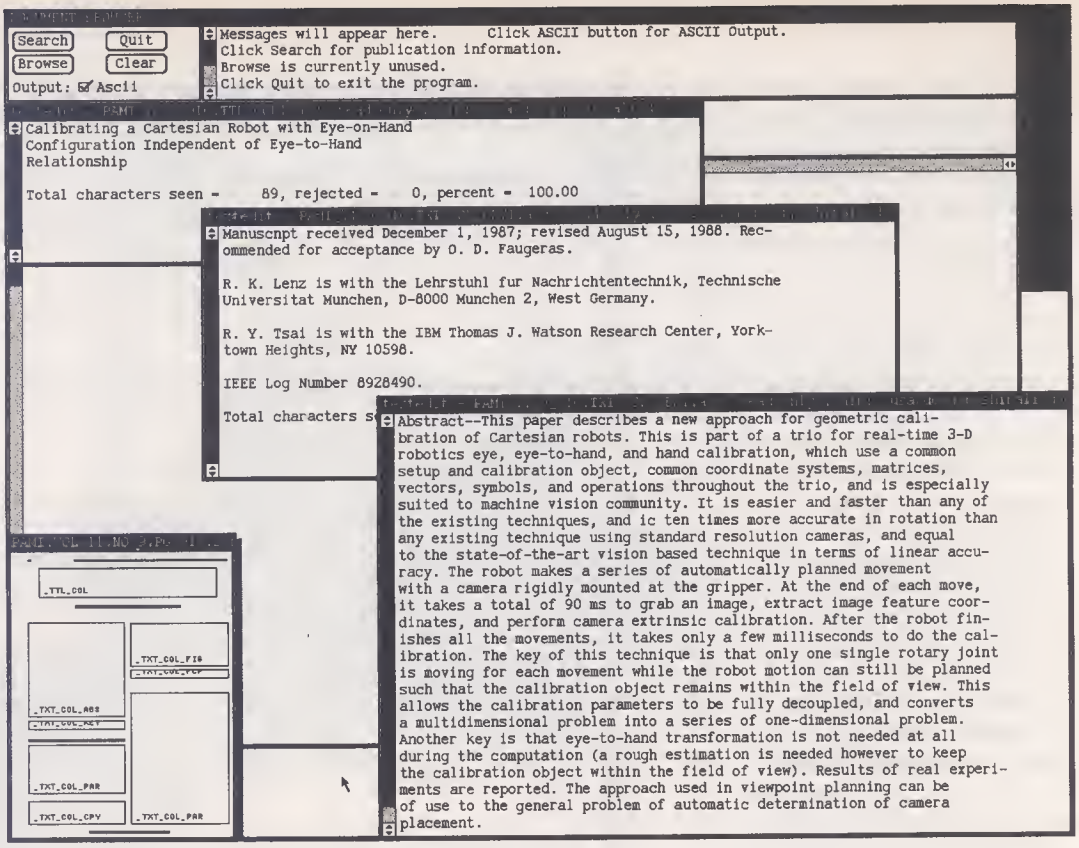


Figure 7. ASCII display of three *IEEE-PAMI* text blocks processed by OmniPage.

ASCII renditions of the title, footnote, and abstract blocks. Note that the word "Manuscript" in the footnote block is incorrectly recognized, as is the word "is" in the seventh line of the abstract. OmniPage records the number of characters and the number of rejects but cannot, of course, calculate the number of substitution errors.

Figure 8 shows the 300-dpi version of the simple line drawing of the sample page. Since the page image is currently available on the IEEE CD-ROM only in binary form, photographs would also be displayed with extreme contrast. The image can be either scrolled or zoomed if its size exceeds the screen size. Figure 9 shows both the image and ASCII displays of a text block — the figure caption.

Experimental results. We have successfully processed 21 photocopied pages from the *IBM Journal of Research and Development* and 20 pages from *IEEE-PAMI*. Since these pages were used for development and improvement of the page grammars, we then randomly selected 12 pages from each journal for an independent test. Nine of the *PAMI* pages were segmented and labeled perfectly. Three had minor mistakes. Of the 12 *IBM Journal* pages, seven were segmented perfectly, three had minor mistakes, and two missed about one quarter of the page. All mistakes could be corrected by simple modifications of the block grammars, but it is clear that several design-and-test cycles would be required for acceptable performance. An interactive step, similar

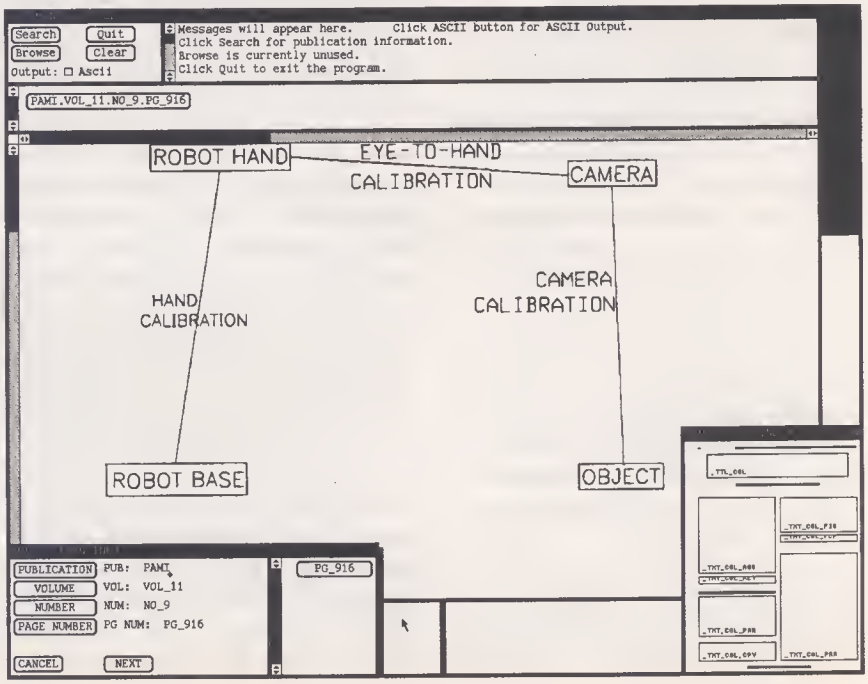


Figure 8. Image display of a figure block. This block was selected by clicking on the topmost block in the right-hand column of the geometric representation (bottom right). If the figure is small enough, it is shown at full 300-dpi resolution on the screen; otherwise it is reduced or scrolled.

to postediting in OCR, could also be invoked when the algorithm fails. Since failures can be flagged by the system itself, the overall throughput would not be greatly affected.

The bulk of the two to three minutes required to process each new page is consumed in recursive profile extraction. The algorithms were coded with little regard to efficiency; for instance, we used shell scripts whenever possible. We have recently implemented a profile extraction algorithm on a 32 × 32-processor DAP (distributed array of processors) computer. Initial comparisons show that the time required to extract the horizontal and vertical profiles of a 2,000 × 3,000-pixel image is reduced to one tenth.

Although an ASCII representation of technical documents is adequate for many purposes, a faithful rendition of the original layout is highly desirable for human access. Not only is this rendition necessary for graphics, equations, and tables whose computer representation is not yet standardized, but also preservation of the original layout and typography enhances legibility compared with OCR output, which preserves only some of this information.

We have demonstrated a prototype version of a system, based on syntactic document analysis and OCR, that can provide useful remote access to stored technical documents. The two aspects that differentiate our system from others are (1) X-Y tree data structures that are particularly suitable for printed matter, and (2) syntactic analysis of image blocks at increasing levels of refinement.

We are now interfacing our prototype browser with the Rensselaer Polytechnic Institute library information system, Infotrax. The system already includes all IEEE indexing data, including abstracts, since 1988. Infotrax yields, for each article of potential interest, the publication title, volume, issue, and page number. Our sample documents are already indexed with the corresponding information.

In addition to its use for information retrieval, we intend to adapt our system to provide input for automated or interactive indexing. Once an article is processed, only ASCII versions of such relevant fields as title, author, and abstract would be forwarded to the indexing station. The body of the text would remain available for subsequent full-text searches, as opposed to searches on selected index terms only.

Longer-term research objectives include developing improved methods for the acquisition of publication-specific knowledge bases, possibly including some form of learning. We are, however, also investigating to what extent documents can be analyzed by using generic typesetting knowledge only. ■

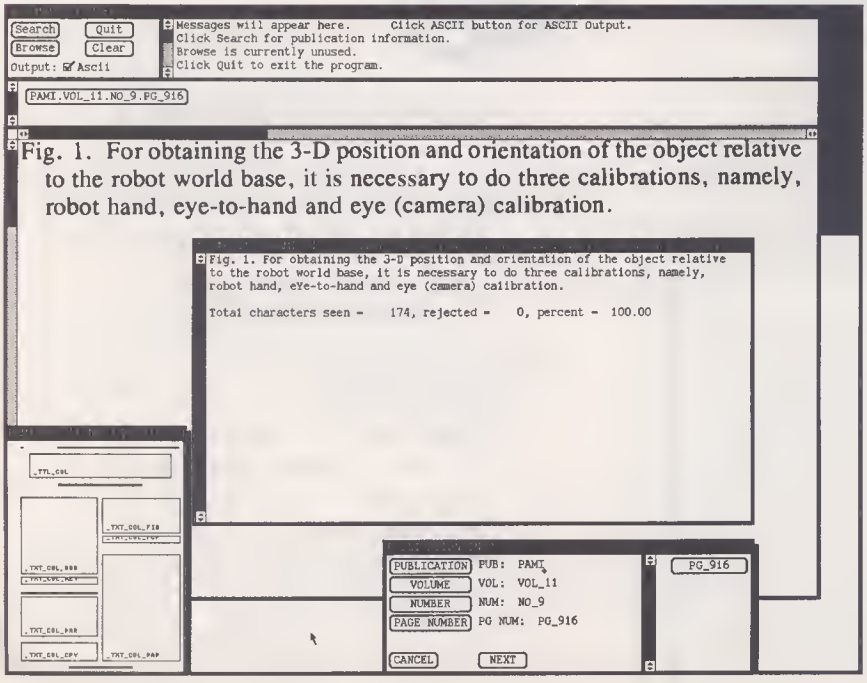


Fig. 1. For obtaining the 3-D position and orientation of the object relative to the robot world base, it is necessary to do three calibrations, namely, robot hand, eye-to-hand and eye (camera) calibration.

Figure 9. Image and ASCII versions of the figure-caption block. The lowercase "y" in "eye-to-hand" in the last line was incorrectly identified as uppercase.

Acknowledgments

Former and current students who have contributed extensively to the project are J. Kanai, E. McCaughrin, R. Sathyamurthy, N. Shirali, and J. Waclawik. Our collaborators at Rensselaer include M. Krishnamoorthy and T. Spencer, and Associate Director of Rensselaer Libraries P. Molholt. We are pleased to thank H.S. Baird (AT&T Bell Laboratories) for document-compression software and E. Annese (Olivetti Corp., Italy) for de-skewing software. The work was supported in part by US West Advanced Technologies and the US Department of Education College Library Technology and Cooperation Grants Program.

References

1. R.G. Casey and K.Y. Wong, "Document-Analysis Systems and Techniques," *Image Analysis Applications*, R. Kasturi and M. M. Trivedi, eds., Marcel Dekker, New York, 1990, pp. 1-36.

2. H.S. Baird, "The Skew Angle of Printed Documents," *Proc. SPSE 40th Conf. Symp. Hybrid Imaging Systems*, SPSE, Springfield, Va., 1987, pp. 14-21.

3. K.R. McConnell, D. Bodson, and R. Schaphorst, *Fax: Digital Facsimile Technology and Applications*, Artech House, Norwood, Mass., 1989, p. 96.

4. *Structured Document Image Analysis*, H.S. Baird, H. Bunke, and K. Yamamoto, eds., Springer-Verlag, 1992, in press.

5. *ICDAR, Proc. First Int'l Conf. Document Analysis and Recognition*, AFCET, Paris, 1991.

6. G. Nagy and S. Seth, "Hierarchical Representation of Optically Scanned Documents," *Proc. Seventh Int'l Conf. Pattern Recognition*, Order No. M545 (microfiche), IEEE CS Press, Los Alamitos, Calif., 1984, pp. 347-349.

7. M. Viswanathan, "Analysis of Scanned Documents — A Syntactic Approach,"

in *Structured Document Image Analysis*, H. Baird, H. Bunke, and K. Yamamoto, eds., Springer-Verlag, 1992, in press.

8. J.-M. De La Beaujardiere, "Well-Established Document Interchange Formats," in *Document Manipulation and Typography*, J.C. van Vliet, ed., Cambridge University Press, 1988, pp. 83-94.

9. R. McLean, *The Thames and Hudson Manual of Typography*, Thames and Hudson, London, 1980.

10. R. Bradford and T. Nartker, "Error Correlation in Contemporary OCR Systems," *Proc. First Int'l Conf. Document Analysis and Recognition*, AFCET, Paris, 1991, pp. 516-523.

11. V. Garza et al., "OCR Product Comparison," *Infoworld*, Oct. 22, 1990, pp. 73-90.

12. L. Grunin, "OCR Software Moves into the Mainstream," *PC Magazine*, Oct. 30, 1990, pp. 299-350.

George Nagy has been professor of computer engineering at Rensselaer Polytechnic Institute since 1985. Before that he was professor of computer science at the University of Nebraska at Lincoln, where he worked on remote sensing applications, geographic information systems, computational geometry, and human-computer interfaces. He also conducted research on various aspects of pattern recognition and OCR at the IBM T.J. Watson Research Center for 10 years. In addition to digitized document analysis and character recognition, his interests include solid modeling, finite-precision spatial computation, and computer vision.

Nagy received the BEng degree in engineering physics and the MEng degree in electrical engineering from McGill University and the PhD degree in electrical engineering (on neural networks) from Cornell University in 1962. He is a senior member of the IEEE, a member of the Computer Society, and a member of ACM.

Sharad Seth is a professor in the Computer Science and Engineering Department at the University of Nebraska at Lincoln. Besides document analysis, his research has been in testing and testable design of microelectronic circuits.

Seth received the BEng degree in 1964 from the University of Jabalpur, India, the MTech degree in 1966 from the Indian Institute of Technology, Kanpur, and the PhD degree in electrical engineering in 1970 from the University of Illinois at Urbana-Champaign. He serves on the editorial boards of *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, and the *Journal of Electronic Testing: Theory and Applications* (*JETTA*). Seth is a senior member of the IEEE, a member of the Computer Socicty, and a member of ACM.

Mahesh Viswanathan is a development staff member with IBM Storage Systems Products Division, San Jose, California. His research interests include printing and document analysis.

Viswanathan received the BSc degree in physics in 1980 from Loyola College, Madras, India, and the BE degree in electrical engineering in 1984 from the Indian Institute of Science, Bangalore. He received the MS degree in electrical engineering from San Diego State University in 1986 and the PhD degree in computer and systems engineering in 1990 from Rensselaer Polytechnic Institute.

Readers can contact George Nagy at Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180; (518) 276-6078; fax (518) 276-6261; e-mail gnagy@mts.rpi.edu.

VICTORIA UNIVERSITY OF TECHNOLOGY

FOOTSCRAY CAMPUS

PO Box 14428, MMC, Melbourne, Vic. Australia 3000
Phone (613) 688 4315. Fax (613) 688 4965

Chair of Computer Science (Level E)

The Department of Computer and Mathematical Sciences wishes to appoint a Senior Academic to the Chair of Computer Science. The Department is a constituent of the Faculty of Science and offers undergraduate and postgraduate courses and masters and doctoral degrees by research. The Department's course and research interests are directed towards the application of mathematical and computer based techniques to industry and business. The major research interests in computer science are in the areas of software engineering, object oriented methodology, artificial intelligence, networks, parallel processing and image processing.

You will be responsible for providing academic, research and professional leadership in Computer Science but the major responsibility of the position will be the development of a distinctive research profile in Computer Science within the Department. Some teaching duties in your area of expertise will also be expected.

A PhD with relevant professional experience in Computer Science is essential. You will also be expected to have a proven record in research with experience in developing and administering research activities.

The appointment will be made on a tenured basis. Salary will be A$77,900 per annum (Level E). Superannuation, travel and removal allowance and temporary housing assistance are available.

For further information contact Associate Professor Ian Murray, Head of Department on (613) 688 4690.

Contact Footscray Campus Human Resources for a position description and forward your written application to the above address (preferably two copies) quoting Reference Number F/CM02 and stating the names and telephone number(s) of three referees by 20th July, 1992. Quote Campus on envelope.

AK18865

ICPR '92

CALL FOR ATTENDANCE

11th International Conference on Pattern Recognition

August 30-September 3, 1992
The Hague, The Netherlands

For further information contact:

ICPR Secretariat
Delft University of Technology
Electrical Engineering Department
PO Box 5031
NL-2600 GA Delft
The Netherlands
Telephone: 31(15)786-052
Facsimile: 31 (15) 622-000
e-mail: icpr@et.tudelft.nl

IEEE First International Workshop on Systems Management

April 14-16, 1993
University of California
Los Angeles, California

Call for Participation

Sponsored by IEEE Computer Society
in cooperation with UCLA and IBM

With the growth of large inter-connected systems, software and hardware techniques for the integrated management of these systems to meet user requirements is of increasing importance. Systems Management techniques need to encompass the diverse operating systems, heterogeneous software and hardware found in many enterprises. This workshop is intended to provide a forum to discuss issues, methodologies and experiences in systems management.

Participants of the workshop are required to submit a 3-5 page position paper (extended abstract, summary, or working paper) on systems management. Topics include (but are not limited to):

- network management
- enterprise management
- software distribution
- backup and recovery
- fault tolerance
- name management
- management protocols
- management services
- management information bases
- knowledge based techniques
- system management application software
- case studies
- standards
- architectures and frameworks
- platform technologies and tools
- application programming interfaces
- security
- monitoring
- performance management and modeling
- policy

Schedule

Position Papers Due - **October 16, 1992**
Notification of Acceptance - **December 7, 1992**
Camera Ready Copy Due - **January 22, 1993**

Submit six copies of the manuscript to:
Dr. Seraphin B. Calo, Workshop Chair
IBM Research Division
30 Saw Mill River Road
Hawthorne, New York 10532
email: calo@watson.ibm.com
phone: (914) 784-7514

Program Committee

Seraphin B. Calo - IBM Research (Workshop Chair)
Wesley Chu - UCLA (co-Chair)
Allan Finkel - IBM Research (co-Chair)
Mike Accetta - Carnegie Mellon University
A. Bouloutas - National Technical University of Athens
Stephen Crocker - Trusted Information Systems
Kathryn DeNitto - Open Software Foundation
Keith Marzullo - Cornell University
George Mouradian - AT&T Bell Labs
Jan Olsson - IBM Sweden
Lisa Phifer - Bellcore
Peter Roden - MIT
Mischa Schwartz - Columbia University
Mark Sylor - Digital Equipment Corporation

THE FOLLOWING INFORMATION IS AVAILABLE:

Contact the Publications Office;
to facilitate handling, please request by number.

- Membership application, student #203, others #202
- Periodicals subscription form for individuals #200
- Periodicals subscription form for organizations #199
- Publications catalog #201
- Compmail electronic mail brochure #194
- Technical committee list/application #197
- Chapters lists, start-up procedures #193
- Student scholarship information #192
- Volunteer leaders/staff directory #196
- IEEE senior member grade application #204
  (requires ten years practice and significant performance in five of those ten)

To check membership status or report a change of address, call the IEEE toll-free number, 1-800-678-4333. Direct all other Computer Society-related questions to the Publications Office.

PURPOSE

The IEEE Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 100,000 members worldwide, and provides a wide range of services to members and nonmembers.

MEMBERSHIP

Members receive the acclaimed monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.

IEEE COMPUTER SOCIETY®

A member society of the
Institute of Electrical and Electronics Engineers, Inc.

PUBLICATIONS AND ACTIVITIES

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, interviews, and product reviews.

Periodicals. The society publishes seven magazines and five research transactions. Refer to membership application or request information as noted at left.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 100 titles every year.

Standards Working Groups. More than 100 of these groups produce IEEE standards used throughout the industrial world.

Technical Committees. More than 30 TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.

Chapters. Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

OMBUDSMAN

Members experiencing problems — magazine delivery, membership status, or unresolved complaints — may write to the ombudsman at the Publications Office.

EXECUTIVE COMMITTEE

President: Bruce D. Shriver*
17 Bethea Drive
Ossining, NY 10562
Phone: (914) 762-3251
Fax: (914) 941-9181

President-Elect: James H. Aylor*
Past President: Duncan H. Lawrie*

VP, Conferences and Tutorials: Barry W. Johnson (1st VP)*
VP, Educational Activities: Raymond E. Miller (2nd VP)*
VP, Membership Activities: Fiorenza C. Albert-Howard*
VP, Press Activities: Yale N. Patt*
VP, Publications: Harold S. Stone*
VP, Standards Activities: Gary Robinson†
VP, Technical Activities: Joseph Boykin†

Secretary: Ronald G. Hoelzeman*
Treasurer: Laurel V. Kaleda†
IEEE Division V Director: Bill D. Carroll†
IEEE Division VIII Director: Helen M. Wood†
Executive Director: T. Michael Elliott†

*voting member of the Board of Governors
†nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 1992:
Alicja I. Ellis, Ronald G. Hoelzeman, Tadao Ichikawa, C.V. Ramamoorthy, Sallie V. Sheppard, Harold Stone, Akihiko Yamada

Term Expiring 1993:
Fiorenza Albert-Howard, Jon T. Butler, Michael C. Mulder, Yale N. Patt, Anneliese von Mayrhauser, Benjamin W. Wah, Ronald Waxman

Term Expiring 1994:
Mario R. Barbacci, Luis-Felipe Cabrera, Wolfgang K. Giloi, Guylaine M. Pollock, John P. Riganati, Ronald D. Williams, Thomas W. Williams

Next Board Meeting
November 20, 1992, 8:30 a.m.
Minneapolis Marriott, Minneapolis, Minn.

SENIOR STAFF

Executive Director: T. Michael Elliott
Publisher: H. True Seaborn
Director, Conferences and Tutorials: Anne Marie Kelly
Director, Finance and Information Services:
Director, Board and Administrative Services: Violet S. Doan
Assistant to the Executive Director: Sandra K. Pfau

COMPUTER SOCIETY OFFICES

Headquarters Office
1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Phone: (202) 371-0101
Fax: (202) 728-9614

Publications Office
10662 Los Vaqueros Cir.
PO Box 3014
Los Alamitos, CA 90720-1264
Membership and General Information:
(714) 821-8380
Publication Orders: (800) 272-6657
Fax: (714) 821-4010

European Office
13, Ave. de L'Aquilon
B-1200 Brussels, Belgium
Phone: 32 (2) 770-21-98
Fax: 32 (2) 770-85-05

Asia/Pacific Office
Ooshima Building
2-19-1 Minami-Aoyama, Minato-ku
Tokyo 107, Japan
Phone: 81 (3) 3408-3118
Fax: 81 (3) 3408-3553

IEEE OFFICERS

President: Merrill W. Buckley, Jr.
President-Elect: Martha Sloan
Past President: Eric E. Sumner
Secretary: Karsten E. Drangeid
Treasurer: Theodore W. Hissey, Jr

VP, Educational Activities: Edward A. Parrish
VP, Professional Activities: Arvid G. Larson
VP, Publication Activities: James T. Cain
VP, Regional Activities: Luis T. Gandia
VP, Standards Activities: Marco W. Migliaro
VP, Technical Activities: Fernando Aldana

An Interpretation System for Land Register Maps

Luca Boatto, Vincenzo Consorti, Monica Del Buono, Silvano Di Zenzo, Vincenzo Eramo, Alessandra Esposito, Francesco Melcarne, Marco Meucci, Andrea Morelli, Marco Mosciatti, Stefano Scarci, and Marco Tucci

IBM Southern Europe Middle East and Africa Scientific and Technical Solution Center

The semantics of land register maps drive this document conversion system. However, its methods of image representation, vectorization, and symbol recognition can be generalized to other classes of line drawings.

Creating a cartographic database often involves the acquisition of huge amounts of data from paper drawings. The acquisition is usually performed with hand-operated digitizing tablets, following procedures that are time-consuming, costly, and error-prone. Effective techniques for the automatic input of drawings into a database are difficult to implement, and the many efforts in this direction over the past 20 years have found limited success. Only recently have substantial advances been achieved in this field.[1-5]

We have developed a system for the automatic acquisition of land register maps. The system converts paper-based documents for the Italian Land Register Authority into digital form for integration into an existing database. Compliance with Land Register Authority standards was a basic design issue. These standards are strictly prescribed, and geometric entities (for example, areas relevant from the viewpoint of land taxation) must be computed within narrow tolerances.

Processing a map begins with its digitization by a scanning device. A key step in our system is the conversion from raster format to *graph representation*, a special binary image format suitable for processing line structures. Subsequent steps include vectorization of the line structures and recognition of the symbols interspersed in the drawing. The system requires operator interaction only to resolve ambiguities and correct errors in the automatic processes. The final result is a set of descriptors for all detected map entities, which is then stored in a database.

This system relies upon, and is driven by, the semantics of land register maps. However, the image representation, vectorization techniques, and optical character recognition subsystem are quite general. The methodology implemented in the system can be generalized to the acquisition of other classes of line drawings.

0018-9162/92/0700-0025$03.00 © 1992 IEEE

Land register cartography

Figure 1 shows an example of a land register map, also called a cadastral map. Cadastral maps describe the geometry of land properties and buildings in a geographical context. They divide territory into a number of polygons, each one representing a piece of land, a building, a street, or a body of water. The scale can range from 1:500 to 1:5,000. A set of adjacent polygons representing land or buildings belonging to the same owner is called a *parcel*. As shown in Figure 1, each parcel is labeled with a unique number. A separate file records cadastral information related to parcels, such as owner, area, and usage.

Italian land register maps are monochrome drawings, in A0 format (100 × 70 centimeters), containing continuous lines, dashed lines, shading patterns, and symbols (alphanumeric characters and cadastral symbols). Strings of characters identify high-level entities such as parcels, streets, and water. These character strings usually do not touch lines. On the other hand, cadastral symbols, which have special meanings, usually do overlap lines.

The Italian Land Register Authority issues a set of guidelines for drawing a cadastral map. These rules form a graphic language and allow the reader to understand the drawing. However, it is not possible to rely entirely on these guidelines for the automatic interpretation of a map, since draftspeople do not always follow them strictly.

About 300,000 paper-based docu-

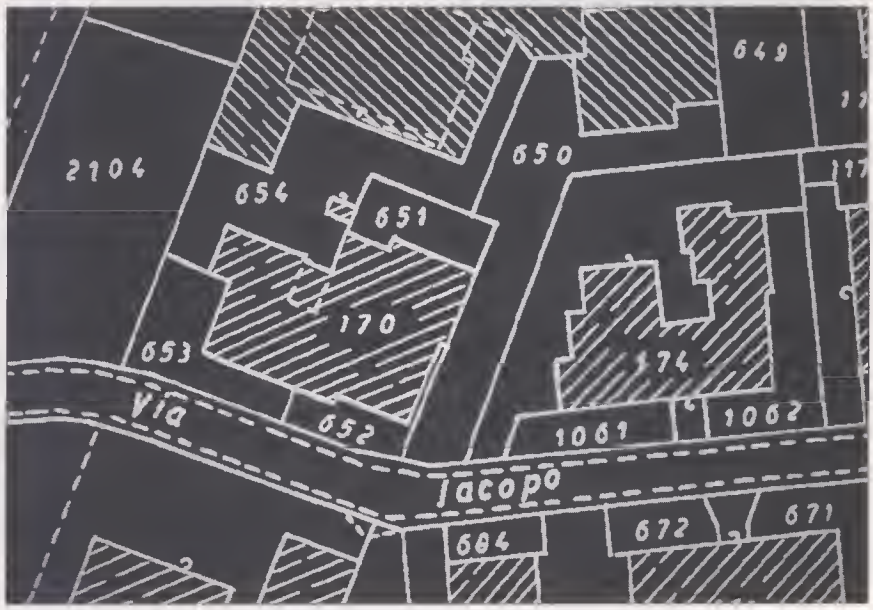


Figure 1. Portion of an Italian Land Register Authority map. Polygons represent pieces of land (unfilled areas), buildings (shaded areas), and streets (labeled by a name). Two adjacent polygons belonging to the same owner are joined by a tilde (~) crossing the common boundary. A set of adjacent pieces of land and buildings belonging to the same owner is called a parcel and labeled by a number placed inside one of the polygons (or alongside, with an arrow pointing to it). For example, parcel 174 consists of a building and two pieces of land. Cadastral symbols (arrow, tilde, bone, etc.) and dashed lines have various meanings, depending on the context.

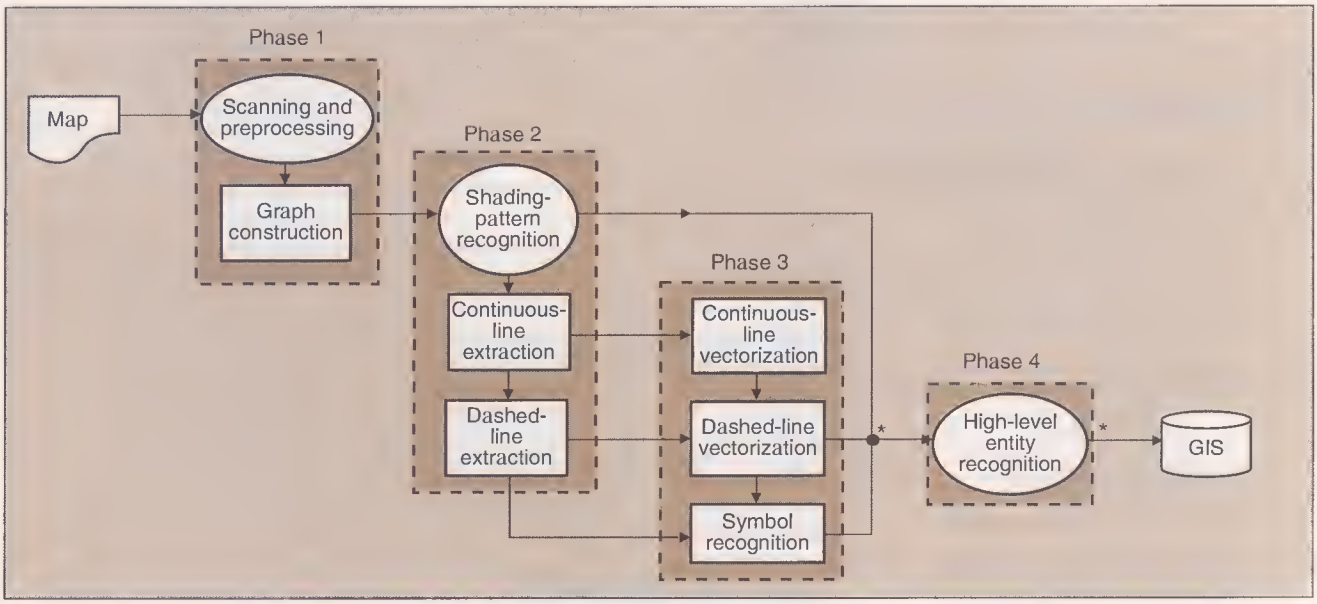


Figure 2. Logical scheme of map interpretation. Boxes and ellipses correspond to the processing steps: Those steps performed automatically by the system are represented by boxes; those requiring operator's interaction are represented by ellipses. Arrows show the dataflow. Asterisks indicate an interactive session, during which the operator checks the output of previous steps.

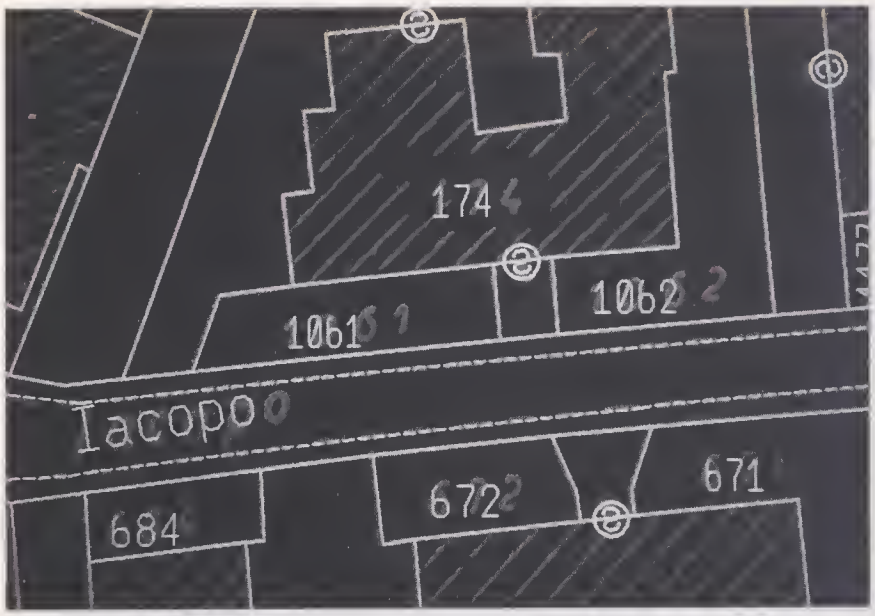


Figure 3. Graphic output of the system. Lines, dashed lines, characters, and cadastral symbols recognized by the system are superimposed on the original image.

ments are being replaced by a computer-based geographic information system. This GIS supports the query, display, and updating of geometric and alphanumeric data describing high-level entities.

System overview

A variety of approaches[6] have been proposed to solve the problems of interpreting line drawings automatically. Text is usually separated from graphics by extracting and classifying connected components in the image.[5,7] This procedure fails, however, when symbols touch or overlap lines — something that occurs often in land register maps. The main problem in recognizing symbols is the variation in their size and orientation. A template-matching approach is inadequate for resolving this problem, but a feature-based approach can work.[2]

In our opinion, the choice of image representation plays a fundamental role in the automatic interpretation of drawings. Accordingly, the first phase in our system generates the graph representation we use in all subsequent processing.

Figure 2 illustrates the phases and steps of the interpretation process. In phase 1, the document is scanned and its graph-encoded binary image is produced. The result is an image decomposed into independent elements, namely, edges and nodes of the graph. In phase 2, each element of the image graph is classified as belonging to one of a set of basic categories such as continuous lines, dashed lines, symbols, and shading patterns.

Phase 3 provides proper descriptions for each labeled image piece, given its category. For example, if an image piece has been recognized as a continuous line, its description will consist of its polygonal approximation by a sequence of vectors. A dashed line would undergo a different vectorization process. Characters and cadastral symbols are also recognized in this phase.

In these first three phases, the system builds a description of the overall geometry of the map or, more specifically, a set of detailed descriptions of the individual geometric entities. System performance in phase 2 is enhanced by relying on the semantics of land register maps, while phases 1 and 3 are largely independent of the class of line drawings being processed.

Finally, in phase 4, the system establishes spatial and semantic relations among geometric entities and also recognizes high-level entities.

Because it is used in legal matters, the cadastral database must have no errors. The operator, therefore, has complete supervision of all decisions made by the system. Operator intervention is required either to resolve ambiguities found by the system during interactive steps or to check the output of fully automatic steps. Additional data validation is performed automatically by checking that the descriptions produced for cadastral entities satisfy logical and geometric constraints.

Figure 3 shows results of interpretation for part of the cadastral map in Figure 1. The remainder of this article describes the main processing steps in Figure 2.

Scanning and preprocessing

This step converts the map into a digitized image suitable for the subsequent processing. The maps are scanned at a sampling rate of 20 points per linear millimeter, with a dynamic range of 256 gray levels. A thresholding operation is then used to obtain a two-level, or binary, image. Each pixel is classified as belonging to the foreground or the background according to whether its gray level is greater or lower than a suitable threshold. The output of this step is a list of *runs* representing the image (see the sidebar titled "Representing the structure of line-like images"). The average cadastral map in this representation occupies about four megabytes of storage.

It is crucial for the thresholded image to reproduce the original as accurately as possible. In particular, the actual width of lines should be preserved everywhere in the image. The threshold is set to the gray level shared by those pixels that differ most in brightness from their neighbors. In fact, these pixels are most likely to be on the boundary between foreground and background.

Thresholding can inject noise into the image. For example, it can create small holes along lines and various irregularities in the figure boundaries. A smoothing process is used to eliminate most of this noise, but some inevitably remains. Consequently, robustness to digitization noise is a basic design requirement for all subsequent processing.

Representing the structure of line-like images

The binary image of a drawing can be represented as a list of vertical or horizontal *runs* (that is, maximal sequences of black pixels in a vertical column or horizontal row, respectively). A vertical run is identified by three integers: its column, the row of the first pixel, and the row of the last pixel. (A horizontal run is identified by its row, the column of the first pixel, and the column of the last pixel.) A subrun is a nonmaximal sequence of black pixels.

The graph construction in our system is based on the run representation.[1] In a *vertical simple graph representation*, the graph is derived using only vertical runs. Figure A shows the image partitioning into edges and nodes for a vertical graph representation. The partitioning is based on the following intuitive observations:

(a) A run adjacent to only one run on each side is likely to be part of a line portion.
(b) A run adjacent to more than one run on a side is likely to be at a crossing point.
(c) A run not adjacent to any run on one side is likely to be at the extreme of a segment.

Such considerations can be formalized and a procedure derived[1] so that each *edge* of the graph is a set of adjacent runs of class (a), while each *node* consists of a single run of class (b) or (c).

This representation can be improved by introducing the notions of crossing area and extreme area as generalizations of the notions of crossing point and extreme point. The notion of areas allows the extension of nodes to include runs previously belonging to edges, so that the lengths of runs in an edge are as uniform as possible (within a certain tolerance due to noise). In Figure B, part (a) highlights the horizontal crossing area and part (b) the vertical crossing area between two lines.

The simple graph representation may fail to associate an edge with a line segment that forms a narrow angle with the direction of runs. The *mixed graph representation* has been introduced to solve this problem. In the mixed graph representation, edges consist of either horizontal or vertical runs, according to the slope of the corresponding line portion, while nodes consist of vertical runs and subruns. In Figure B, part (c) illustrates a mixed graph representation.

The first step in constructing a mixed graph representation is to build both horizontal and vertical simple graph representations, as shown in Figure B, parts (a) and (b). Second, edges are built as sets of adjacent short runs that belong neither to crossing areas nor to extreme areas in the simple representations. (We define a vertical and horizontal run as *conjugate* if they cross, and we define a run as *short* if it is strictly shorter than all of its conjugates.) The remaining pieces of the image, encoded as lists of vertical runs and subruns, are the nodes of the mixed graph (Figure B, part (c)).

The shape of each node is then refined by a heuristic procedure that attempts to minimize the area of the node and to maximize the lengths of the connected edges. (This technique often requires splitting runs into subruns.) Finally, edges undergo a polygonal approximation by generating additional nodes so that each resulting edge corresponds to an approximately straight portion of line.

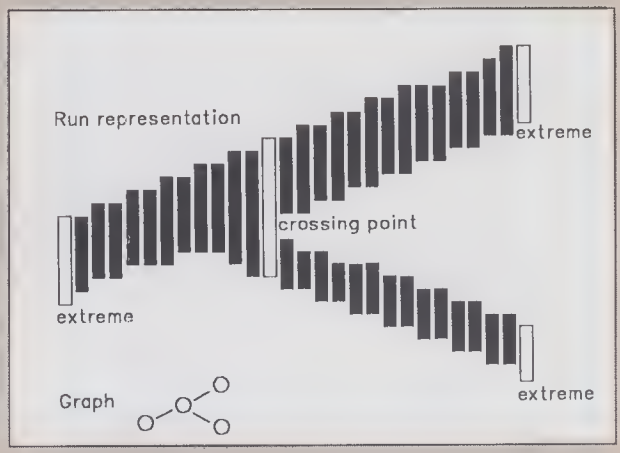


Figure A. First step in the construction of a vertical simple graph representation. Nodes of the graph represent crossing points and extreme points, and correspond to single runs (highlighted in the figure). Edges correspond to remaining sets of adjacent runs.

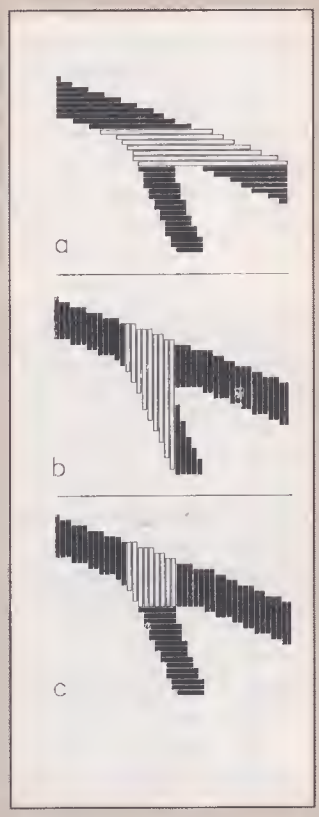


Figure B. Representations of the intersection of two lines. In a horizontal simple graph representation (a) and a vertical simple graph representation (b), the crossing area (corresponding to a node in the graph) is highlighted. In a mixed graph representation (c), each edge corresponds to a set of either vertical or horizontal adjacent runs, according to its slope, and the node is represented by vertical runs and subruns.

Reference

1. S. Di Zenzo and A. Morelli, "A Useful Image Representation," *Proc. Fifth Int'l Conf. Image Analysis and Processing*, World Scientific Publishing, Singapore, 1989.

Graph construction

The drawing interpretation algorithms that we describe here are based on the special image representation that we call *graph representation*. This format is especially convenient for line-like images because it decomposes the line structure into *edges* and *nodes* that formalize the intuitive notions of "lines" and "crossing points between lines." During graph construction, the binary image is partitioned into connected sets of foreground pixels, each corresponding to a graph edge or a graph node. We also use the terms edge and node to mean the image pieces associated with the graph elements.

Our system uses the graph representation for several purposes:

- To detect basic elements of the image (for example, line segments, symbols, and shading patterns) by recognizing corresponding patterns in the graph. (This makes the search for possibly rotated patterns an affordable task, whereas it is almost prohibitively expensive on the raster image.)
- To partition the image into its basic elements (such as segments, symbols, and shaded areas) by recognizing corresponding patterns in the graph.
- To vectorize line segments.
- To help image editing during interactive sessions. (The operator is allowed to pick individual edges and nodes as atomic elements of the image.)

Shading-pattern recognition

In Italian Land Register Authority maps, areas corresponding to buildings are shaded with broken parallel hatching lines spaced 0.5 to 0.7 millimeters apart and forming an angle, preferably 45 degrees, with the perimeter. It is crucial to recognize buildings before the raster-to-vector conversion. Only perimeters have to be vectorized. Hatching lines should be removed from the image graph because they carry no additional information and they introduce noise in the vectorization process. This task is not trivial, because hatching lines touch the perimeter and often overlap symbols or other lines internal to the perimeter.

In detecting a building, the system searches the image graph for subgraphs typically belonging to buildings. Figure 4 gives an example. Nodes 1, 2, and 3 within the chain of collinear edges A, B, C, and D (part of the perimeter) are connected respectively to edges E, F, and G (part of the hatching lines). The edges E, F, and G have no further connections in the graph. When such a comb-like structure is found, the system assumes that a building exists and uses heuristics to search the image graph locally for its perimeter. If the search fails to find a closed polygon, the operator is asked either to reject the candidate building or to interactively help the system identify the perimeter. Finally, all nodes and edges internal to the perimeter, which are recognized as parts of hatching lines, are removed from the image graph.

Continuous-line extraction and vectorization

A component-labeling algorithm is used to derive the connected components of the image, which are then classified as "line" or "sign" based on their radius. (Radius is defined as the maximum distance of the component's pixels from the centroid.) This classification process identifies isolated dashes and symbols.

The system then searches the graph of "line" components to extract symbols overlapping lines. Their subgraphs are usually composed of chains or cycles of short edges, thick edges, or edges for which one of the two endpoints is a terminal node (that is, not connected to other edges). The search locates such small subgraphs and changes the labels of their elements from "line" to "sign." The remaining components of the image graph labeled as "lines" are routed to the vectorization process.

In most drawing analysis systems, the raster-to-vector conversion of a line structure is achieved by thinning the image, thus reducing thick lines to linear sequences of pixels, and then finding an approximating polygonal line. In the latter process, approximation algorithms operate on linearly ordered sets of pixels, which requires each stage of the process to know the next pixel to be approximated.

The graph representation avoids the problems of thinning,[8] since the edges of the image graph are linearly ordered

Figure 4. Mixed graph representation of a building in a cadastral map. The subgraph identified by edges A, B, C, D, E, F, and G typically belongs to a building. (Buildings are shaded by broken parallel hatching lines forming preferably 45-degree angles with the perimeter.) Once such a structure is found in the image graph, the chain of collinear edges A, B, C, and D is used as the starting point for a local search to identify the perimeter of the building.

sets of runs. Various polygonal approximation algorithms[9,10] can be modified to work with sequences of runs. In our system, the approximation actually takes place at an early stage, during the construction of the mixed graph representation (see the sidebar on representing line-like images). The approximating elements are rhomboids, rather than ideal segments of zero width. The rhomboids can reflect the actual width of the line being approximated.

The vectorization process relies on the graph representation. First, sequences of collinear edges are searched according to the following "fitting condition": A straight line exists such that (1) the maximum distance between the line and the middle points of the runs of the edges is lower than a certain threshold and (2) the average value of this distance is lower than a threshold. Then vertices are computed as intersections of such lines. Vectors are derived from vertices. When more than two edges converge into a node, the corresponding lines are modified to create a single intersection point. However, if the modified lines violate the fitting condition, new vectors — not associated with any edge — are introduced to connect those intersection points that cannot be merged.

This technique satisfies the Land Register Authority's strict requirements (0.4-mm tolerance). Even higher accuracy has been reached. In the maps converted for the LRA thus far, all vectors were placed inside the raster image, near the medial axis, and no false vectors were generated.

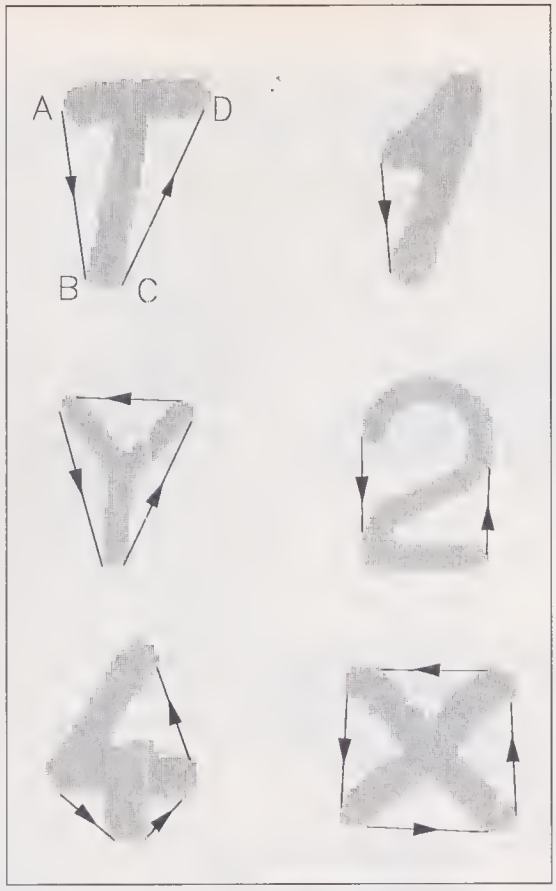


Figure 5. Examples of lid configurations. Lids can be used to derive reliable features for symbol recognition.

Dashed-line extraction and vectorization

Dashes are selected from the set of signs based on the analysis of geometric properties and the recognition of typical patterns in the image graph (for example, node-edge-node). The pool of candidate dashes is then searched for sequences of dashes in which the spacing of adjacent items is lower than a threshold T_D and the angle formed by adjacent items is lower than a threshold T_A. Such a process extracts sequences forming dashed lines that may not be straight. Dashed lines are subsequently approximated by sequences of vectors.

This process, however, is not adequate for the recognition of dashed lines in all possible situations. For example, at the intersection of two dashed lines, different patterns can occur: X, T, L, +, etc. Such patterns are generally labeled as signs, but they are not considered during dashed-line extraction, since they are not recognized as candidate dashes. Hence, they are separately recognized on the basis of their graph structure, and are included in dashed lines if their edges, considered as separate dashes, belong to already identified portions of dashed lines.

After dashed-line extraction, all remaining signs (including unused candidate dashes) are labeled as "symbols."

Symbol recognition

Characters and cadastral symbols are recognized based on geometrical and morphological properties called *features*. Symbols are mostly handprinted, and are arbitrarily rotated and scaled. Features must therefore be invariant under shifts, rotations, and scaling, and robust to distortions and noise. Some reliable features are derived from *lids* (the vectors **AB** and **CD** are the lids of the symbol T in Figure 5) and *sides* (as known from the geometry of plane figures). The patterns formed by lids and sides convey much information about the shape of a figure. Figure 5 shows some examples of possible lid configurations.

With reference to Figure 5, the following is a very simplified set of requirements that a symbol should satisfy for consideration as a candidate T: (1) There are two lids, (2) neither lid is twice as long as the other, and (3) the angle between **AB** and **CD** is greater than 110 degrees.

More generally, each class of symbols has a set of requirements that a shape should satisfy to be considered as a candidate for that class. Each requirement is so formulated that it is satisfied with probability 1 by every shape actually belonging to the class. This strategy allows adding as many requirements as needed for actual discrimination without performance degradation.

Di Zenzo et al.[11] give a detailed description of the optical character recognition system we use.

Many of the symbols found in a map are part of a string. In such cases, the context provides useful information about orientation, size, syntax, etc., which the system uses to resolve ambiguities (for example, 6 versus 9 or O versus 0). The system uses proximity and collinearity criteria to identify strings.

High-level entity recognition

The output of the previous processing steps is a low-level description of the drawing — except for buildings (see previous discussion under "Shading-pattern recognition"). Segments are described in a *vector graph;* characters and cadastral symbols have been recognized. This representation completely replaces the raster image and is the

input for the next processing step, which recognizes the high-level entities of the map such as parcels, streets, and bodies of water.

A cadastral entity is composed of segments and symbols drawn according to specific guidelines. A key point in the design of this part of the interpretation process is the choice of appropriate representations of such guidelines. Our choices were driven by the objective of breaking down the task into a set of well-known problems in graph theory and operational research.

As an example, we describe the technique for identifying cadastral parcels. A parcel is drawn as a set of adjacent polygons, labeled with a number placed inside (when possible) or close to one of the polygons. Two polygons belong to the same parcel if a tilde (~) is drawn on the common edge (see parcel 174 in Figure 1). The first step of parcel recognition locates all the minimal cycles of the vector graph (that is, polygons are identified). Then a new graph is built in which the polygons are the nodes and an edge is created between two nodes if the corresponding polygons are connected by a tilde. The connected components of this graph are the candidate parcels. To find actual cadastral parcels, the procedure looks for an optimal assignment of the set of recognized numbers to the pool of candidate parcels.

The system has been implemented on two IBM platforms. Performance results for Land Register Authority maps are reported in the sidebar titled "System performance."

The graph image representation is a major advantage of our system for interpreting land register maps. It provides a formal description of the image's topological and metrical properties, which has proved useful in solving many problems in the automatic interpretation of line drawings.

The system successfully handles real maps that often contain formal errors and semantic ambiguities. It is fault tolerant and robust to noise. Moreover, its basis in quite general concepts and methodologies makes it suitable for the interpretation of other kinds of line drawings. In fact, this is the focus of our current work. Initial experiments on technological networks and engineering drawings show promising results. ■

System performance

The system consists of a set of programs that communicate through files: The output of one step is the input to the next one. This pipeline structure ensures high software modularity. Furthermore, it allows a batch of maps to be processed simultaneously, thus avoiding operator inactivity during fully automatic steps. Two platforms are currently available: (1) S/370 or S/390 architecture running VM/CMS, equipped with IBM 5080 graphic stations, and (2) IBM RS/6000 workstation running AIX.

The most relevant system-performance measures are elapsed time and operator's intervention time for complete processing of a map sample. However, the accuracy of automatic steps is a key factor, since improving accuracy in these steps can reduce the requirements for interactive operations.

Table A reports the results of shading-pattern recognition (a partially automatic step). Table B reports the results of character recognition (a fully automatic step). Table C shows the average amount of operator's time needed to produce the final database. The results were obtained over a 40-map sample. (On average, a map contains about 600 parcels, 130 buildings, 1,800 characters, and 700 cadastral symbols.) Manual conversion, using digitizing tablets and alphanumeric keyboards, required an average of about 20 hours of operator's time per sample — or 5.7 times more operator's time than the system.

Table A. Results of shading-pattern recognition over a 40-map sample. Both global performance of the step and details of operator intervention are reported. "False buildings" are those parts of the image graph that the system interpreted wrongly as buildings. "Perimeters found" refers to detected buildings.

Description	Number (percentage)
Total buildings in the maps	7,181
Buildings detected	
Automatically	6,564 (91.4)
With operator intervention	617 (8.6)
False buildings detected	57
Perimeters found	
Automatically	6,350 (96.7)
With operator intervention	214 (3.3)
Total	6,564

Table B. Results of automatic character recognition over a 40-map sample.

Characters in the Maps	Number (percentage)
Correctly recognized	66,387 (92.47)
Unrecognized	4,927 (6.84)
Misrecognized	497 (0.69)
Total	71,796

Table C. Average elapsed times needed to process 100 × 70-cm cadastral map on an IBM 4381 processor. Results were obtained over a 40-map sample. Interactive steps account for operator's intervention time.

Description	Time
Phase 1: Preprocessing and graph construction	17 min.
Phases 2 / 3: Construction of low-level geometric description	2 hrs. 39 min.
Phase 4: Recognition of high-level logical entities	32 min.
Interactive steps	2 hrs. 59 min.
Automatic steps	29 min.
Total elapsed time	3 hrs. 28 min.

Acknowledgments

We gratefully acknowledge the encouragement of Herbert Freeman and George Nagy. We also thank Aldo Spirito for his important contribution to our work and Roberto Cremonini for his comments on an earlier draft.

References

1. M. Karima, K.S. Sadhal, and T.O. McNeil, "From Paper Drawings to Computer-Aided Design," *IEEE Computer Graphics and Applications*, Vol. 5, No. 2, 1985, pp. 27-39.

2. M. Ejiri et al., "Automatic Recognition of Engineering Drawings and Maps," *Image Analysis Applications*, R. Kasturi and M.M. Trivedi, eds., Marcel Dekker, New York, 1990.

3. S. Suzuki and T. Yamada, "Maris: Map Recognition Input System," *Pattern Recognition*, Vol. 23, No. 8, 1990, pp. 919-933.

4. M.T. Musavi et al., "A Vision-Based Method to Automate Map Processing," *Pattern Recognition*, Vol. 21, No. 4, 1988, pp. 319-326.

5. S. Bow and R. Kasturi, "A Graphics-Recognition System for Interpretation of Line Drawings," *Image Analysis Applications*, R. Kasturi and M.M. Trivedi, eds., Marcel Dekker, New York, 1990.

6. H. Freeman, "Computer Processing of Line-Drawing Images," *Computing Surveys*, Vol. 6, No. 1, Mar. 1974, pp. 57-97.

7. L.A. Fletcher and R. Kasturi, "A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, Nov. 1988, pp. 910-918.

8. B. Jang and R.T. Chin, "Analysis of Thinning Algorithms Using Mathematical Morphology," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 6, June 1990, pp. 541-551.

9. J. Sklansky and V. Gonzalez, "Fast Polygonal Approximation of Digitized Curves," *Pattern Recognition*, Vol. 12, No. 5, Aug. 1980, pp. 327-331.

10. K. Wall and P. Danielsson, "A Fast Sequential Method for Polygonal Approximation of Digitized Curves," *Computer Vision, Graphics, and Image Processing*, Vol. 28, No. 2, Nov. 1984, pp. 220-227.

11. S. Di Zenzo et al., "Recognizing Hand-printed Characters of Any Size, Position, and Orientation," to be published in *IBM J. Research and Development*, Vol. 36, No. 2, 1992.

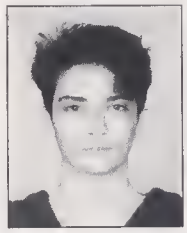
Luca Boatto is a reseàrcher at the IBM Southern Europe Middle East and Africa (SEMEA) Scientific and Technical Solution Center. His research interests include automatic understanding of cadastral maps, engineering drawings, and maps representing technological networks. Boatto was a student fellow at Telettra Digital Signal Processing Laboratory in Rieti and received his degree in electrical engineering from the University of Rome, La Sapienza.

Vincenzo Consorti is a researcher at the IBM SEMEA Scientific and Technical Solution Center. Until 1988, he was with Telettra Digital Signal Processing Laboratory in Chieti, where he worked on electromagnetic compatibility and the theory and design of digital filters. His main research interests involve image recognition and understanding. Consorti received his degree in electrical engineering from the University of Pisa.

Monica Del Buono is a researcher at the IBM SEMEA Scientific and Technical Solution Center. Her research interests include optical character recognition and document analysis. She previously worked in the area of document handling and office automation. Del Buono received her degree in mathematics from the University of Rome, La Sapienza.

Silvano Di Zenzo is director of research at the IBM SEMEA Scientific and Technical Solution Center. He has been assistant professor of mathematics at the University of Genoa, an IBM systems engineer, and manager of an IBM industry project in image processing. His technical interests include machine vision and pattern recognition. Di Zenzo received his degree in electrical engineering from the University of Genoa. He is coeditor of *Image and Vision Computing*.

Vincenzo Eramo is a senior researcher at the IBM SEMEA Scientific and Technical Solution Center, which he joined in 1987. His technical interests include image processing, pattern recognition, and technical drawing understanding. Eramo received his degree in physics from the University of Bologna.



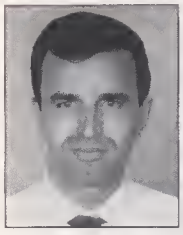
Alessandra Esposito is a researcher at the IBM SEMEA Scientific and Technical Solution Center. She is a member of the Image Processing Group currently involved in the technical drawings processing project—particularly in pattern recognition and image understanding. Esposito received her degree in electrical engineering from the University of Naples, Frederico II.

Francesco Melcarne is a researcher at the IBM SEMEA Scientific and Technical Solution Center. He is a member of the Image Processing Group involved in the technical drawings processing project. Until 1990, he worked at the Institute of Systems Analysis and Computer Science of the Italian National Council of Research on a project concerning the use of artificial intelligence techniques for traffic control. Melcarne received his degree in electrical engineering from the University of Rome, La Sapienza.

Marco Meucci is a system engineer in the IBM SEMEA Geographical Information Systems Solutions Group. His technical interests are in the research and development of image understanding and user interfaces. Meucci received his degree in mathematics at the University of Rome, La Sapienza.

Andrea Morelli is a researcher at the IBM SEMEA Scientific and Technical Solution Center. His research interests involve image recognition and understanding. Morelli received his degree in physics from the University of Milano.

Marco Mosciatti is a system engineer in the IBM SEMEA GIS Solutions Group. Until 1987 he worked as a consulting engineering in the Mechanical Engineering Department of the University of Rome. His research interests are in image understanding and user interfaces. Mosciatti received his degree in mechanical and industrial engineering from the University of Rome, La Sapienza.



Stefano Scarci is manager of engineering applications at the IBM SEMEA Scientific and Technical Solution Center. Until 1991, he was manager of image and knowledge processing, and responsible for the map interpretation project. He has worked on several research projects in signal processing and pattern recognition, both at the IBM SEMEA Scientific and Technical Solution Center and at the IBM T.J. Watson Research Center in the US. Scarci received his degree in electrical engineering from the University of Naples, Frederico II. He is a member of the IEEE.

Marco Tucci is a member of the Image Processing Group at the IBM SEMEA Scientific and Technical Solution Center. Until 1990, he was a researcher with the Computer Science Department of the University of Rome. His research interests include artificial intelligence, software engineering, pattern recognition, and image understanding. Tucci received his degree in electrical engineering from the University of Rome, La Sapienza.

Readers can contact Vincenzo Consorti at IBM SEMEA Scientific and Technical Solution Center, Viale Oceano Pacifico 171, 00144 Rome, Italy; e-mail consorti@romesc.vnet.ibm.com.

Great Work. Great Living.
IBM Burlington, Vermont.

One of the world's most advanced semiconductor operations is what you'll find at IBM's major development and manufacturing facility in Burlington, where continued business growth is matched by a superb living environment. We now have outstanding career opportunities for engineers with the specialized computer skills to make significant impact on RISC microprocessor development.

Logic Design

Responsible for definition, logic design and verification of high performance RISC microprocessors. To qualify, you must possess a BSEE or higher, with an emphasis on Computer Engineering, and be capable of carrying logic design through to physical chip design stage. Minimum of 3 years in logic/chip, CMOS and VLSI design required. RISC experience is key. Background in microprocessor and multiprocessor design desirable.

Circuit Design

Will design CMOS circuitry for RISC-based microprocessor functions. Includes custom SRAM cache design, complex logic dataflow circuitry, random logic, IO, clocking and other circuitry in custom microprocessor layouts. Requires BSEE or higher with emphasis on Computer Engineering or Circuit Design. Ability to design complex CMOS or Bi CMOS circuits and perform circuit analysis and verification is essential, along with minimum of 3 years circuit design experience in industry. CMOS, VLSI, digital circuit design is a prerequisite.

Physical Design

Responsible for CMOS VLSI chip physical design of RISC microprocessor in advanced CMOS technology. Includes using state-of-the-art CAD tools to perform chip layout, wiring and chip timing analysis. A BSEE or higher, with emphasis on Computer Engineering or Circuit Design, is essential, along with at least 3 years of physical design experience in industry. RISC and CMOS, VLSI design experience (chip layout/wiring) necessary. Background in microprocessor design desirable.

Located between Lake Champlain and Vermont's Green Mountains, Burlington offers year round recreation and open space. Unspoiled beauty, affordable housing and a sense of community come together here. This is life at its most enjoyable; technology at its best.

IBM offers salaries commensurate with qualifications and a comprehensive benefit package. For confidential consideration, please send your resume, indicating area of interest, to: **IBM Corporation, Professional Recruiting, 1000 River Street, Essex Junction, VT 05452.**

IBM®

An equal opportunity employer.

Postal Address Block Location in Real Time

Paul W. Palumbo, Sargur N. Srihari, Jung Soh, Ramalingam Sridhar, and Victor Demjanenko
State University of New York at Buffalo

A postal automation system locates destination address blocks on letter mail pieces with a high success rate. Pipelining and multiprocessor techniques achieve real-time processing speed.

In 1990, some 162 billion pieces of mail were received and delivered by the US Postal Service, and the volume is expected to reach 200 billion annual pieces by 1995. Manually sorting mail using ZIP (zone improvement plan) codes requires time-consuming labor; to reduce costs, the postal service is now relying more on automated sorting. (See the "Postal automation" sidebar for details.)

As part of the automated sorting process, the address recognition unit (ARU) locates and reads the destination address. The address block location (ABL) subsystem of the ARU uses the digital image of the entire piece of mail to determine the location of blocks of text that may contain the destination address. The ARU then presents the digital subimage corresponding to the destination address block (DAB) to the address interpretation subsystem, which segments the block into lines and words, locates the candidate ZIP code and other "words," and "reads" each word to determine the destination code (for example, ZIP + 4 code). This article focuses on the development of an ABL subsystem with improved performance for the next-generation US Postal Service ARU.

An ABL subsystem has been under development at the USPS Center of Excellence for Document Analysis and Recognition (CEDAR) since 1985. This subsystem (hereafter referred to as a system) uses an image of the entire mail piece to determine the coordinates of the minimum bounding rectangle enclosing the destination address block. For each candidate DAB, the ABL system determines the line segmentation, global orientation (0, 90, 180, or 270 degrees), block skew, an indication of whether the address appears to be handwritten or machine printed, and a value indicating the degree of confidence that the block actually contains the destination address.

Initially, the CEDAR ABL system was based on a blackboard data structure invoking many specialized image processing and block analysis tools using a rule-based system.[1] The major emphasis of this early (non-real-time) system was on ABL performance rather than processing throughput. All image processing operations were written in C, and the rule-based system was written in Lisp. The processing time required for the system was seven to eight minutes for letters and longer for magazines and parcels.

0018-9162/92/0700-0034$03.00 © 1992 IEEE

A real-time ABL system requires a processing throughput of between 1/18th and 1/9th of a second, depending on mail piece length and transport belt speed. The average throughput should be 1/13th of a second. Because many of the original algorithms were difficult to implement under real-time constraints, we redesigned them to work within the constraints or removed them from the system. For example, the blackboard system had nine specialized feature-extraction tools that would require significant real-time hardware. The system we describe in this article, however, uses only two feature-extraction tools that are relatively easy to implement in real time. In addition, we replaced the rule-based system with a more simplistic evidence combination approach.

Glossary

ABL — Address block location
ADTH — Adaptive thresholding tool
BAT — Block analysis tool
BLCS — Block-splitting tool
CEDAR — USPS Center of Excellence for Document Analysis and Recognition
COMB — Combination tool
COVF — Consistency verification tool
DAB — Destination address block
EVHP — Evidence combination tool
HEUR — Spatial heuristics tool
HSEG — Handwriting segmentation tool
HSEG1 — Column blanker and connected component locator
HWMP — Handwriting/machine-print discrimination tool
LAYO — Block layout tool
LOCA — Block location tool
LSEG — Text line segmentation tool
MSEG — Machine-printed segmentation tool
MSEG1 — Connected component locator
OCR — Optical character recognition
POST — Postprocessing tool
PSEG — Postcard segmentation tool
SIZE — Block size tool
USPS — United States Postal Service
ZIP — Zone improvement plan
ZIPM — ZIP code merging tool

Postal automation

To sort letter mail automatically, the US Postal Service has been using optical character recognition (OCR) machines since the 1970s. Postal OCR systems can handle up to 45,000 pieces per hour and automatically sort roughly 60 percent of the mail (mostly machine printed). Mail pieces are encoded with a five- or nine-digit ZIP code in a bar-code format in the lower right corner of the mail piece. Future processing then requires simply reading the bar code rather than reinterpreting the mail piece.

Figure A shows the major components of a postal OCR system designed by the US Postal Service Center of Excellence for Document Analysis and Recognition. Postal workers place mail pieces in a feeder mechanism that presents one mail piece at a time to the main transport belt. The gap between each mail piece is determined by the feeder to be approximately 3.5 inches. The transport belt moves the mail pieces at 120 inches per second past an image scanner, through a three-second delay, past the bar-code sprayer, and finally into an appropriate sorting bin. The scanner captures a 300-pixel-per-inch gray-scale image using a time-delay integration charge-coupled device array. The three-second delay corresponds to the processing time allocated for locating and reading the destination address so the correct code can be sprayed as a bar code.

During this delay an address recognition unit locates and interprets the destination address block. The unit includes subsystems for address block location and address interpretation. The unit uses postal directories to extend the printed information to sort the mail piece to nine digits, even when the address doesn't explicitly contain a "ZIP+4" code.

The goal of the USPS-supported research program is to improve OCR performance by developing new address-recognition and image-lift units to be used with the existing transport mechanism. The 1995 goal is *delivery point encoding* of 90 percent of all mail with a machine-printed destination address and 50 percent of all mail pieces with a handwritten destination address. Delivery point encoding is a much finer level of sort than the current OCR systems can perform (typically an 11-digit code rather than a five- or nine-digit ZIP code), and it requires large national address databases. To achieve these demanding goals, the performance of both the address block location (ABL) and address interpretation subsystems needs to be significantly improved.

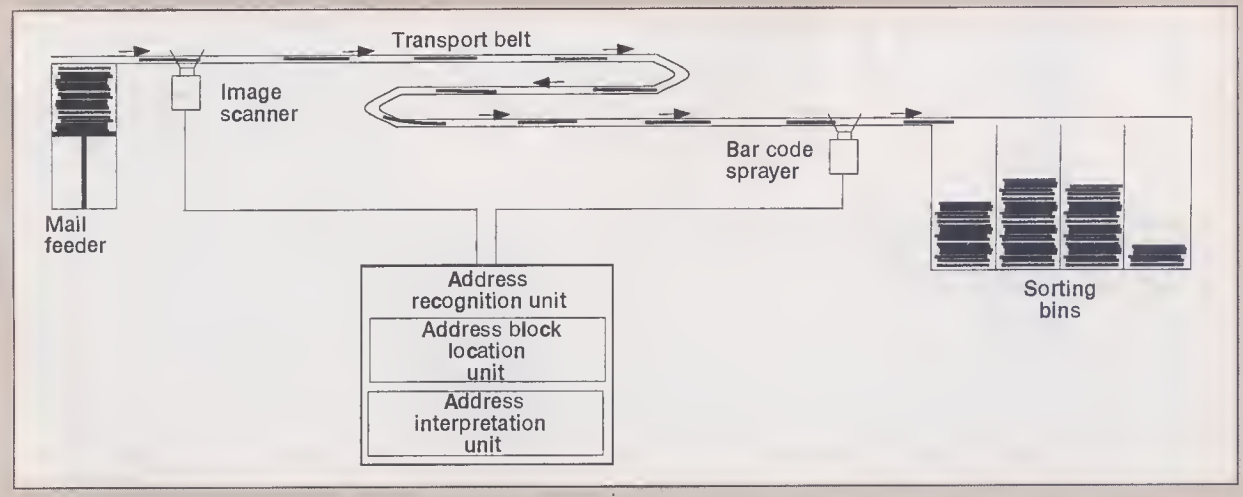


Figure A. Postal optical character recognition unit design.

System description

The current system (shown in Figure 1 and described in Table 1) is based on bottom-up or data-driven processing. It extracts primitive information from the image and groups the information into possible DABs. This framework is very conducive to real-time operation. The data presented to the real-time ABL system is an image quantized at the relatively low resolution of 100 points per inch. We selected this low resolution to reduce image processing time, but even at this resolution, a standard-size letter (4 1/2 by 9 1/2 inches) produces approximately 400,000 points of data (pixels).

This large amount of data is analyzed using fast, specialized image processing hardware. The hardware extracts interesting features from the image for further analysis with software running on general-purpose processors. The processors perform the actual grouping (segmentation), classify the print method of the text within the segmented block, and select the destination address block from the blocks segmented by the system. Pipelined image processing hardware along with multiprocessing of symbolic data provides the speed, minimum latency, and flexibility required for the system.

Since image processing hardware typically is faster than general-purpose processors, a pool of processors sustains the real-time throughput. At any given time, one processor services data transferred to it from image processing hardware and other processors are at various stages of locating a candidate DAB on different mail pieces. This multiprocessor approach permits easy addition and subtraction of processors within the pool and relatively easy software development. To control this hybrid system, a system controller determines the next processor to be used from the pool and communicates the results to the address interpretation subsystem.

Instead of analyzing all the segmented blocks at the same time, the system independently ranks the candidate blocks from each of four primary segmentation tools. The four ranked lists are subsequently combined into one list, and the top-ranked block becomes the system choice for the DAB. This approach yields better performance than analyzing all the blocks together, especially when the system attempts to find blocks that satisfy spatial relations (for example, the spatial relation between the return address and the destination address).

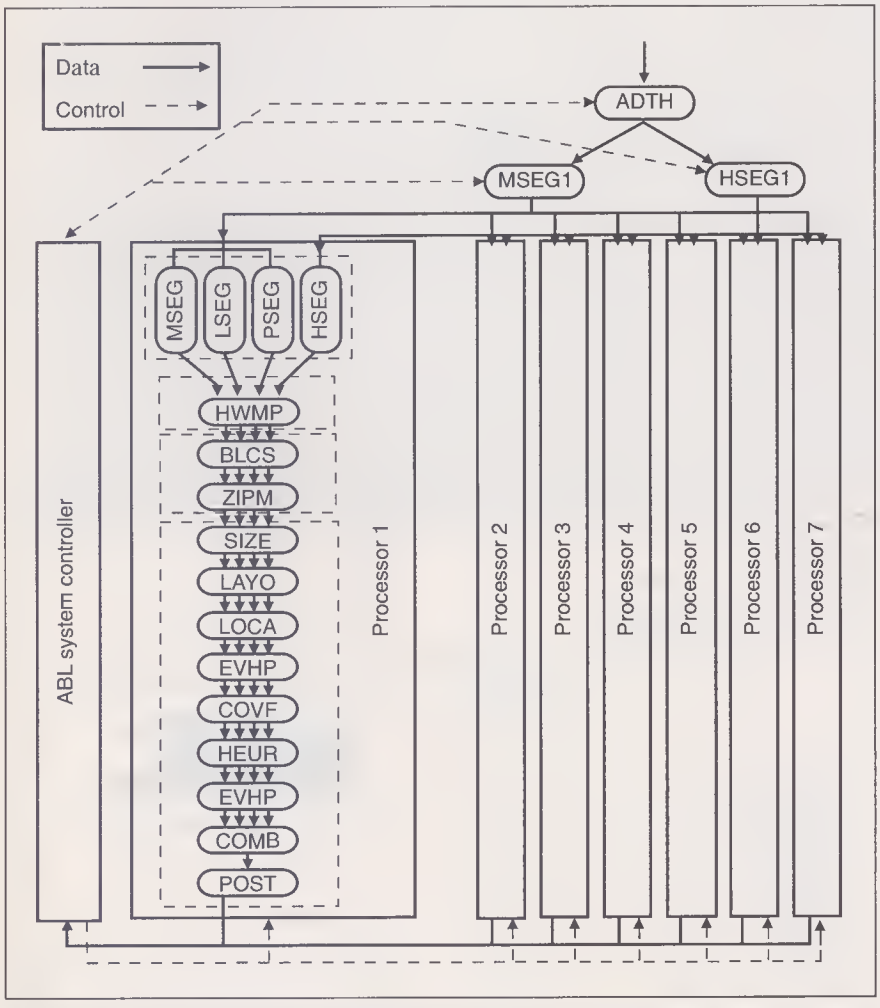


Figure 1. System design for real-time address block location.

Image processing. The image processing operations required by the CEDAR real-time ABL system are thresholding and component location. Specialized image processing hardware[2] minimizes processing time and latency (see the "MaxBus processing" sidebar) and includes off-the-shelf and custom-designed modules.

Adaptive thresholding. The adaptive thresholding tool[3] (ADTH in Figure 1) converts a gray-level image into a two-valued (binary) image. A wide variety of image quality is encountered in this application — for example, image quality is affected by colored and textured envelopes, poor illumination, and digitizing noise. We determined empirically that a locally adaptive thresholding method would perform best. The algorithm convolves the input image with an 8×8 contrast mask and thresholds the convolution output using a lookup table to produce a binary image. This tool is presently implemented with two commercially available boards manufactured by Datacube.

Connected-component location. After the image has been converted to binary form, information is extracted using connected-component analysis. A connected component is a set of all foreground pixels immediately adjacent to each other. Typically, in a machine-printed DAB under ideal digitizing conditions, each alphanumeric character is a separate connected component. The extent of each connected component is then used by the symbolic processing portion of the system to locate candidate DABs. The problem of locating connected components has been widely

Table 1. Description of tools in the real-time ABL system.

Category	Tool	Description
Image processing	ADTH	Adaptive thresholding to convert a gray-level image into a binary image using local contrast.
	MSEG1	Locates connected components from a binary image corresponding to machine-generated characters and some handwritten characters.
	HSEG1	Locates connected components from a binary image corresponding to handwritten characters and some machine-generated characters.
Primary segmentation	MSEG	Bottom-up segmenter that groups components from a binary image into primarily machine-generated words, lines, and blocks.
	LSEG	Bottom-up segmenter that groups components from a binary image first into lines, then into blocks.
	PSEG	Bottom-up postcard segmentation tool.
	HSEG	Bottom-up segmenter that groups components from a binary filtered image into primarily hand-generated words, lines, and blocks.
Print method	HWMP	Classifies a segmented block as being printed either by hand or by machine.
Segmentation refinement	ZIPM	Merges a small block to the lower right of a destination address candidate when it appears to be a ZIP code.
	BLCS	Splits a large machine-generated or handwritten text block into several smaller text blocks.
Block analysis	SIZE	Uses block features (aspect ratio, length, height, number of text lines, and number of components) to classify how likely a block is to be a destination address.
	LAYO	Uses characteristic layout of text lines in destination address to estimate the likelihood of a block being the destination address.
	LOCA	Uses the location of a block to determine the likelihood of this block being the destination address, return address, or postage.
	EVHP	Pools together the evidence generated by various tools and generates labeling hypotheses.
	COVF	Verifies the consistency of labeling hypotheses among neighboring blocks by using spatial relations.
	HEUR	Uses spatial heuristics or rules of thumb to choose the destination address from a list of candidates.
	COMB	Selects the best candidate block from among the highly ranked blocks.
	POST	Adjusts the segmentation of the top-ranked candidate destination address block, if needed.

MaxBus processing

The MaxBus, developed by Datacube, is a point-to-point bus that transfers an image between image processing boards. The bus permits modular design of the image processing stages required for each specific application. Each pixel is processed in scan-line order, synchronized to a 104-nanosecond pixel clock. Horizontal and vertical synchronization signals indicate the start of each scan line and image frame, respectively. At every pixel clock, the bus clocks an image pixel in and out of the image processing hardware in a one-dimensional raster data stream. The bus implements two-dimensional image operations, such as image convolution, by buffering several scan lines of image data to create a two-dimensional window for processing.

Figure B shows how a 3 × 3-pixel window is produced from a one-dimensional stream of pixels. Pixels are clocked into the window, starting with the northwest element, and move to the north and northeast elements before being buffered in a line buffer for nearly a complete scan line. A pixel traverses through the other elements of the window before being discarded when it leaves the southeast element. This serpentine path generates a two-dimensional operation using a one-dimensional data stream.

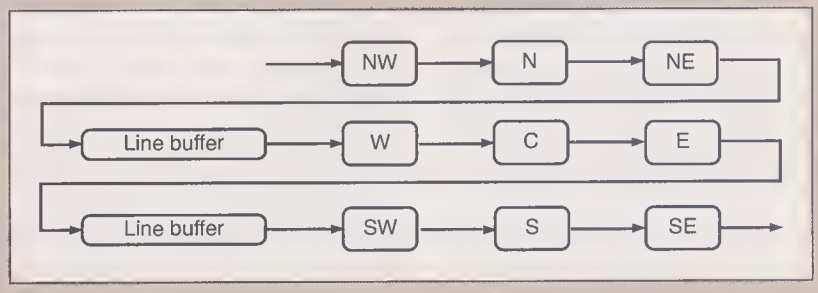


Figure B. Two-dimensional processing using MaxBus.

explored.[4,5] We needed an algorithm that used only one pass at real-time rates to conform to the MaxBus specification. The CEDAR custom algorithm[6] is a fast, minimal latency, one-pass design that accepts MaxBus input. The hardware for the connected-component locator comprises three custom boards and a custom VLSI content-addressable memory chip.[7]

Two separate system modules locate connected components: MSEG1 and HSEG1. Although they work best on extracting features from machine-printed and handwritten addresses, respectively, both modules process all images, irrespective of their actual print method. The primary difference between these two tools is that MSEG1 locates connected components in the binary image while HSEG1 locates connected components in the binary image *after* vertically "whiting-out" every 32nd column. Thereby, HSEG1 attempts to break up cursive script characters to reduce the size of the components but without using any underlying image information. More complex filtering algorithms have been tested, but this simple method produces acceptable results and is easy to implement in hardware. The connected-component data produced by these two tools is sent to the general-purpose processors using a direct memory access interface over the VMEbus.

Symbolic processing. After the connected-component location tools convert the image into symbolic data, all subsequent processing is performed on general-purpose processors. We programmed these processors with software written in C using Wind River Systems' VxWorks real-time operating system.[8] The symbolic processing tools are divided into four main functions: component grouping, print method classification, segmentation refinement, and analysis of the segmented blocks.

Component grouping. The component grouping tools cluster connected components into text lines and text blocks. The four bottom-up primary segmentation approaches use different algorithms and produce different segmentations of the same mail piece. The segmentation tools are designed for machine-printed addresses (MSEG), formation of better address text lines (LSEG), postcards (PSEG), and handwritten addresses (HSEG). This four-segmentation scheme makes the system more robust and easier to tune. The size of components and distance between them are used to group the components into blocks and lines. All components linked together in this manner are grouped into the same line or block candidate.

Print method classification. To help the downstream OCR system recognize characters, a handwriting/machine-print discrimination tool (HWMP) determines the print method of each block segmented by the ABL system. The algorithm used in the real-time ABL system relies on the frequency of different component heights in the block. It assumes a block with many different heights is handwritten and a block with uniform component heights is machine printed.

Segmentation refinement. The blocks from the four primary segmentation tools undergo a split-and-merge segmentation refinement to further improve their quality.

Sometimes, it is difficult for the primary segmentation tools to avoid producing larger blocks than desired for the DAB because the address is close to extraneous matter (such as a date, an attention line, advertising text, or other markings or graphics on the mail piece). The block-splitting tool (BLCS) splits these large machine-printed or handwritten blocks, which are usually too tall compared with their length to be the DAB. In general, the tool determines where to split a block on the basis of lines with uniform line height, distance between lines, and line-end alignment.

Machine-printed address blocks often have extra space between the city/state names and the ZIP code or between the city name and the state/ZIP code. During the connected-component grouping, this excessive space results in text being excluded from the main body of the address. The ZIP code merging tool (ZIPM) recovers from this inevitable exclusion of the ZIP code by using relaxed block size tests to identify the syntax of most possible DABs. Similarly, it chooses possible state and ZIP code blocks according to block size and number of components inside the block. For each pair of DAB and ZIP code candidates, the tool checks a set of conditions, including relative locations and the ratio of candidate block heights. If the conditions are satisfied, the tool unifies the two blocks and creates evidence for the new block being the DAB. We designed the tool primarily to merge machine-printed ZIP codes but improved it to merge handwritten ZIP codes as well.

Block analysis. A block analysis tool (BAT) is one of a collection of relatively simple tools that analyze the candidate blocks produced by the four primary segmentation sources (MSEG, LSEG, PSEG, and HSEG) and the two secondary segmentation sources (BLCS and ZIPM) to produce a minimum bounding rectangle enclosing the potential address, an associated confidence value, and other properties if the address is found. If the system cannot make a good guess at the address, it produces a reject code. Also, each segmented block is interpreted at four orientations (0, 90, 180, and 270 degrees) to determine the global orientation of the mail piece. A BAT's objectives vary:

(1) extraction of individual block features (size, layout, and location),
(2) application of heuristics based on spatial relationships between blocks (consistency verification and spatial heuristics), and
(3) integration of block features into block labeling hypotheses (evidence combination).

The block size tool (SIZE) uses statistics on DABs derived from a mail statistics database: the mean and standard deviation of the length, height, aspect ratio, number of lines, and number of characters. The tool uses a size score for each segmented block. The closer a block's features are to its statistical mean, the higher the block's score and the more evidence assigned for the block being the DAB.

Besides the size features, DAB text lines have some typical layout characteristics. The block layout tool (LAYO) uses three such characteristics: left-end alignment, line-height variation, and block density. Address lines often have their left ends aligned. Many machine-printed addresses are exactly aligned, but handwritten addresses are often more or less aligned. The LAYO measures the line-height variation by computing the ratio of the standard deviation to the mean of line heights. This ratio is usually relatively small for address blocks. The block density is the average ratio of line length to block

length. The higher this measure, the denser the block and the more likely it is to contain an address. The LAYO evaluates these measures for candidate blocks and generates supporting evidence if left justification, low line-height variation, or high block density is found.

The system uses the location of a candidate block on the mail piece to estimate the likelihood of the block being the DAB, return address block, and postage. In letter mail, DABs are likely to be in the bottom two thirds of the mail piece, return address blocks in the upper left corner, and postage in the upper right corner. The block location tool (LOCA) uses the position of a block and a mail statistics database to compute a location value that compares the block location with "typical" blocks. It uses the location value to associate evidence with a block in support of its being the DAB as well as in support of its being the return address block and postage.

The evidence combination tool (EVHP) integrates pieces of evidence generated for a block into a single block-labeling hypothesis. An evidence frame consists of an attribute (for example, location, layout, or ZIP merging) and four confidence values corresponding to the likelihoods of the block being the DAB, return address block, postage, and extraneous matter. The tool combines these multiple pieces of evidence into a single labeling hypothesis that also has these four confidence values. By sorting the hypothesis confidence values for the DAB, the tool develops a ranked list of candidate DABs. The EVHP is invoked twice, once after the extraction of individual block features and again after the operations of the tools that use spatial relations between blocks. The method to combine pieces of evidence is based on the Dempster-Shafer theory,[9] which uses a commutative combination rule, making the order of combined evidence frames irrelevant.

Some spatial relationships generally hold between blocks on a mail piece. For example, the DAB is below and to the right of the return address block. For each pair of a DAB candidate b_d and a return address block candidate b_r, the consistency verification tool (COVF) checks whether the labeling hypotheses for b_d and b_r are consistent with the spatial relation between them. If the actual spatial relation agrees with a typical DAB-return address block spatial relation, then the DAB confidence of b_d and the return address block confidence of b_r are increased. If the relation disagrees (for example, the DAB candidate b_d is above and to the left of the return address block candidate b_r), then the corresponding DAB and return address block confidence values are decreased. Thus, the consistency verification tool strengthens or weakens confidence values in hypotheses based on the interblock relations.

The spatial heuristics tool (HEUR) examines the spatial relation of the top-ranked block and other blocks with high confidence values for their being the DAB. The goal is to detect an incorrect top choice and select an alternate block. This is achieved by applying a rule of thumb to a pair consisting of top-ranked b_1 and a highly ranked block b_h. For instance, if b_1 is above b_h, then b_1 might be the return address block, while b_h might be the DAB. The tool generates corresponding evidence pieces to reflect this situation. If b_1 is to the right of b_h and has bigger characters inside, then b_1 may be an advertising text block and b_h the DAB. Thus, this tool applies simple rules obtained from experience with mail piece images to cases where multiple blocks have high confidence values for being the DAB.

Each of the four segmentation streams produces an independently derived ranked list of candidate blocks. The combination tool (COMB) merges these lists into one. At present, it selects the top candidate for the destination address from each list, but a more elaborate scheme could use the top, say, three choices from each list. The tool ranks the properties of these top choices and determines the choice most likely to be the destination address. If the top choices from different streams produce nearly the same segmented region, they are treated as one candidate.

The postprocessing tool (POST) attempts to correct for minor segmentation problems in the top-ranked candidate DAB. It removes extra material printed to the left of the address, such as the words "To" and "Pay to the order of." This tool also uses top-down knowledge to correct any possible minor segmentation problems.

System controller. The CEDAR real-time ABL system is a mixture of both specialized pipelined hardware and multiprocessors. A system controller implemented on a dedicated processor controls image processing boards and routes information in and out of general-purpose processors. System controller software configures the image processors by setting the direct memory access address where the connected components will be written. A scheduling algorithm determines routing, selecting the next processor to use and time-out conditions to maintain a constant flow of images through the system. For example, if all the processors are in use when the next image is about to come through, the scheduling algorithm must free a processor to accept the symbolic information from the incoming mail piece or discard the mail piece.

The scheduling algorithm interrupts the job that has been processing for the longest time. However, if this job has not been processing very long (even though it is the oldest one in the system), the incoming mail piece is discarded or "sunk." Sinking prevents processing from being continually interrupted without any significant image-receiving processing time. Mail pieces that are sunk or interrupted by the scheduling algorithm produce no DAB candidates and require manual sorting. The scheduling algorithm minimizes the number of time-outs and provides an effective system throughput rate.

Implementation

The system described here has been integrated and runs at rates of up to 10.9 mail pieces per second. All the image processing hardware, including the custom boards, has been designed, implemented, and tested. The symbolic software runs on four 20-megahertz Sparcengines using VxWorks and controlled by another Sparcengine. In this section, we describe the timing analysis and location performance of the system. All performance evaluations used 2,000 letter mail images (63 percent machine-printed destination addresses and 37 percent handwritten destination addresses) that could not be sorted to nine-digit accuracy by current postal OCR systems.

Timing analysis. The image processing hardware time, based on the Datacube MaxBus standard, is directly proportional to the image size processed (see the MaxBus sidebar). Each pixel

Table 2. Image and symbolic processing times (in seconds).

	Moments		Extremes		Percentile		
	Mean	Standard Deviation	Minimum	Maximum	90	95	99
Image	0.041	0.009	0.022	0.059	0.047	0.047	0.059
Symbolic	0.169	0.084	0.046	1.106	0.246	0.277	0.470
Total	0.210	0.085	0.086	1.133	0.288	0.320	0.504

Table 3. Time-outs at various image-throughput rates (pieces per second).

	Throughput Rates			
	8.2	9.8	10.5	10.9
Time-outs	5	7	15	22

can be processed within 104 nanoseconds, with additional pixel delays for horizontal and vertical image blanking. In the worst case (the maximum letter size), a 6 1/8 × 11 1/2-inch area corresponds to 704,950 pixels when digitized at 100 points per inch and requires only 0.076 scconds including overhead. The latency time through the image processing tools is roughly 10 scan lines (1,228 microseconds). Table 2 shows the image processing time for the 2,000 test images. In general, these times are relatively short. The symbolic processing time of the implementation is also relatively small (see Table 2). We are constantly improving the symbolic processing algorithms to decrease the processing time even further.

We simulated the system design using the SES/Workbench design specification, modeling, and simulation package. The simulation provided information about design effectiveness which helped guide the implementation. Using this simulation, we could relatively easily determine the effect on the system of different scheduling algorithms and timing requirements.

Table 3 shows the number of image time-outs at various image-throughput rates (ranging from 8.2 to 10.9 pieces per second). Images are presented to the ABL system using a real-time disk subsystem, which limits throughput rates to 10.9 pieces per second. The subsystem will eventually be replaced by an image lift under development. Faster processors would permit more complex algorithms and a decrease in the number of system processors.

Location performance. The segmentation results with the test images show that the line-based segmenter (LSEG) is the best of the four primary segmentation tools. As Table 4 shows, it correctly segments (groups) 82.7 percent of the images. The machine-printed address segmenter (MSEG) has a relatively low location performance for handwritten DABs but performs well on machine-printed DABs. Three of the primary segmentation tools (MSEG, LSEG, and HSEG) are used on all images, but the PSEG, BLCS, and ZIPM tools are used only on blocks that meet these tools' preconditions. The overall segmentation performance — that is, at least one of these six segmentation tools correctly locates the DAB — is 96.7 percent. For images in which at least one tool located the DAB, print-method discrimination was correct 96.8 percent of the time.

Once the blocks have been correctly segmented, the BATs must select one as the system's choice for the DAB. As Table 4 shows, the system correctly segments and locates the DAB for 89.0 percent of these images (90.0 percent for machine-printed DABs, 87.2 percent for handwritten DABs). The strict guidelines for this grading were developed by the USPS. In most failures, the system locates most of the destination address but not all of it (for example, a ZIP code is missing), or the system includes extra material such as part of a cancellation or meter mark.

Figure 2 shows how the real-time ABL produces a top-ranked DAB. Figure 2a shows the connected components located by the MSEG1 tool. (HSEG1 output is not shown.) Figures 2b through 2e show the primary segmentation results from MSEG, HSEG, LSEG, and PSEG, respectively. The darker shaded blocks in Figures 2b through 2e represent the top-ranked block for the BATs using that segmentation stream. Figure 2f shows the system's overall top-ranked block (with line segmentation), derived from the top-ranked blocks from the individual segmentation streams. In this example, two segmentation tools (HSEG and PSEG) could correctly segment and rank the destination address.

The CEDAR real-time address block location system determines candidates for the location of the destination address from a scanned mail piece image. With 20-MHz Sparc processors, the average time per mail piece for the combined hardware and software system components is 0.210 seconds. The system locates 89.0 percent of the addresses as the top choice. Recent developments in the system include the use of a top-down segmenta-

Table 4. System performance (success and failure shown in percent).

Grade	Segmentation							Overall Performance
	MSEG	LSEG	HSEG	PSEG	BLCS	ZIPM	Combined	
Success	62.5	82.7	76.7	55.2	49.4	91.6	96.7	89.0
Failure	37.5	17.3	23.3	44.8	50.6	8.4	3.3	11.0

tion tool, address syntax analysis using only connected component data, and improvements to the segmentation refinement routines. This has increased top choice performance to 91.4 percent. Additional improvements are planned, including the use of character recognition and the use of glassine window-detection information; together, these improvements should enhance ABL performance to meet USPS goals. ■

Acknowledgments

We thank Rajesh Dixit, Chandima Edirisinghe, David Fishback, Parag Gokhale, Jeon-Man Park, Yong-Chul Shin, and Madhusudhan Srinivasan of CEDAR as well

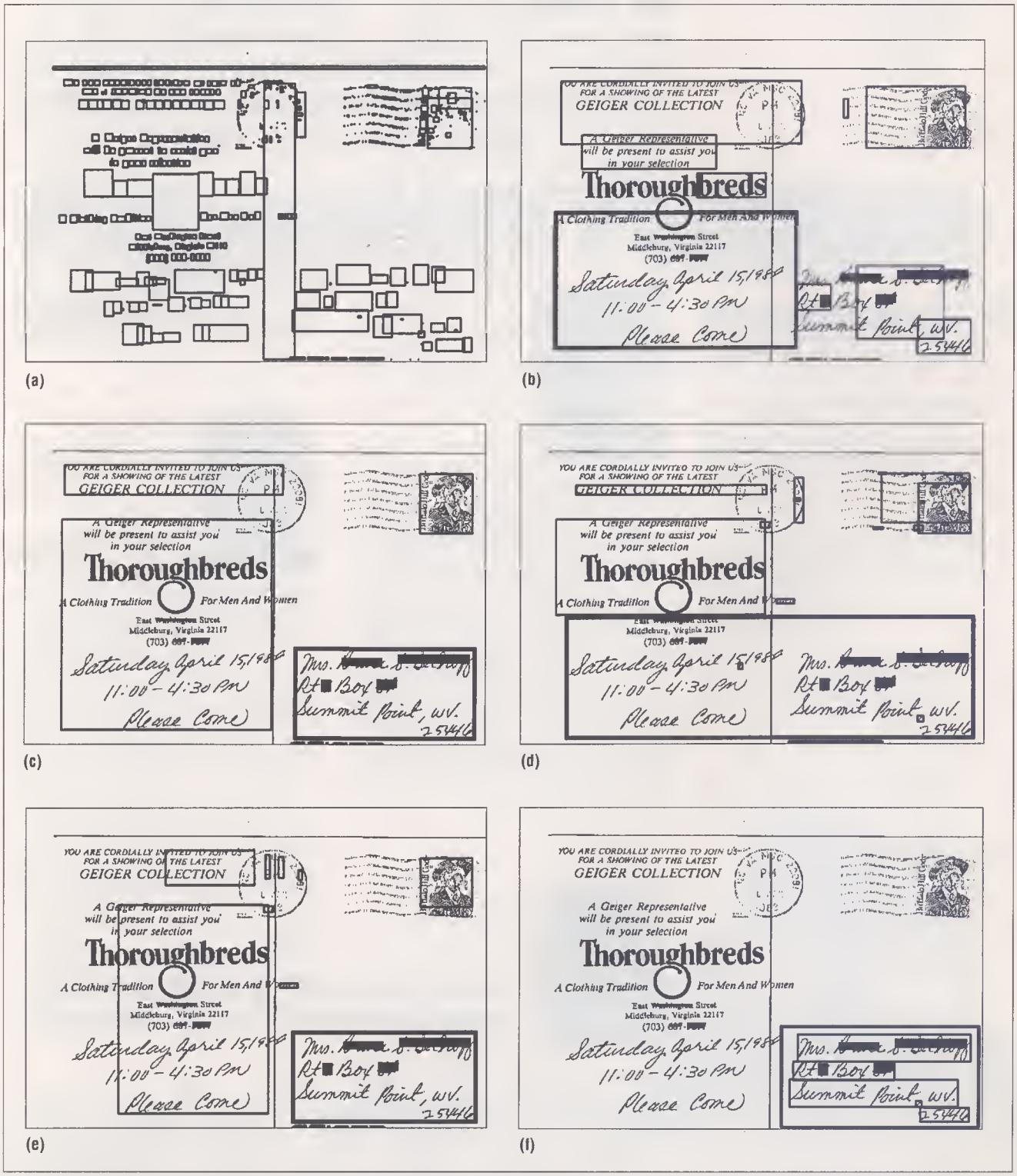


Figure 2. Address block location. MSEG1 extracts connected components (a) from the binary image. The components are grouped into possible blocks by the four primary segmentation tools MSEG (b), HSEG (c), LSEG (d), and PSEG (e). Then, the system produces the top-ranked destination address (f).

as Charlie Stenard of AT&T Holmdel for their assistance on this project. Steve Chahal and Carl O'Connor from the USPS Office of Advanced Technology also provided valuable suggestions for the development of CEDAR's ABL system.

This work was supported by the USPS Technology Resource Department under Task Order 104230-88D-2267.

References

1. C.H. Wang and S.N. Srihari, "A Framework for Object Recognition in a Visually Complex Environment and its Application to Location Address Blocks on Mail Pieces," *Int'l J. Computer Vision*, Vol. 2, No. 2, Sept. 1988, pp. 125-151.

2. J. Walker, "Pipeline Processing for Machine Vision," *Electronic Image East*, BIS CAP Int'l, Boston, 1989, pp. 579-584.

3. P.W. Palumbo, P. Swaminathan, and S.N. Srihari, "Document Image Binarization: Comparison of Techniques," *Proc. SPIE Symp. Digital Image Processing*, SPIE, Bellingham, Wash., Vol. 697, 1986, pp. 278-285.

4. R. Lumia, L. Shapiro, and O. Zuniga, "A New Connected-Components Algorithm for Virtual Memory Computers," *Computer Vision, Graphics, and Image Processing*, Vol. 22, No. 2, May 1983, pp. 287-300.

5. C. Ronse and P.A. Devijver, *Connected Components in Binary Images: The Detection Problem*, John Wiley, Letchworth, UK, 1984.

6. V. Demjanenko et al., "Real-Time Connected Component Analysis for Address Block Location," *Proc. USPS Advanced Technology Conf.*, USPS, Washington, D.C., 1990, pp. 1,059-1,071.

7. Y.C. Shin et al., "A Special-Purpose Content Addressable Memory Chip for Real-Time Image Processing," *IEEE J. Solid State Circuits*, Vol. 27, No. 5, May 1992, pp. 737-744.

8. J. Fiddler, E. Stromberg, and D.N. Wilner, "Software Considerations for Real-Time RISC," *Digest of Papers Compcon Spring 90, 35th IEEE Computer Soc. Int'l Conf.*, IEEE CS Press, Los Alamitos, Calif., Order No. 2028, 1990, pp. 274-277.

9. J.A. Barnett, "Computational Methods for a Mathematical Theory of Evidence," *Seventh Int'l Joint Conf. Artificial Intelligence*, Morgan Kaufmann, San Mateo, Calif., 1981, pp. 868-875.

Paul W. Palumbo is the project manager for the CEDAR Real-Time Address Block Location Project at the State University of New York at Buffalo. He has worked on postal research at SUNY Buffalo since 1984. His research interests are image processing, segmentation, and parallel processing.

Palumbo received his BA and MS degrees at SUNY Buffalo in 1983 and 1985, respectively. He is a member of the IEEE, the IEEE Computer Society, and the ACM.

Sargur N. Srihari is director of the USPS Center of Excellence for Document Analysis and Recognition and holds the Pattern Recognition Professorship in the Computer Science Department at SUNY Buffalo.

Srihari received his BSc in physics and mathematics from Bangalore University in 1967, his BEng in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1970, and his MS and PhD in computer and information science from Ohio State University in 1971 and 1976, respectively. He is a senior member of the IEEE and a member of the IEEE Computer Society, the American Association for Artificial Intelligence, the Pattern Recognition Society, and the ACM. He received a New York State/United University Professions Excellence Award in 1991.

Jung Soh is a research assistant in the Department of Computer Science at SUNY Buffalo. His research interests are in real-time control structures for knowledge-based vision systems.

Soh received a BS in electrical engineering from the University of Wisconsin-Madison and an MS in computer science from SUNY Buffalo.

Ramalingam Sridhar is an assistant professor of electrical and computer engineering at SUNY Buffalo. His research interests include language-directed computer architecture, asynchronous processor design, real-time computer architecture, special-purpose processor architectures, and VLSI design. He is involved in the development of direct-execution processors and real-time postal address recognition systems.

Sridhar received his BE in electrical engineering from the University of Madras, India, in 1980 and his MS and PhD in electrical and computer engineering from Washington State University in 1983 and 1987, respectively. He is a member of the IEEE Computer Society, and the ACM.

Victor Demjanenko is an assistant professor of electrical and computer engineering at SUNY Buffalo. He is also president of Tree Technologies Corp., Buffalo, which specializes in product development and serves as consultant to other organizations. His research interests include real-time image processing, custom computer architectures, network-operating systems, and microprocessor-based control and diagnostic systems. He is involved in the development of real-time postal address reading machines and in the integration of vibrational signal acquisition and diagnostic signal processing.

Demjanenko received his BS, MS, and PhD in electrical engineering from SUNY Buffalo in 1980, 1981, and 1983, respectively. He is a member of the IEEE Computer Society.

Readers can contact Palumbo at the Center of Excellence for Document Analysis and Recognition, 226 Bell Hall, State University of New York, Buffalo, NY 14260. His e-mail address is palumbo@cs.buffalo.edu.

Association for Computing Machinery SIGPLAN

CONFERENCE ON OBJECT-ORIENTED PROGRAMMING SYSTEMS, LANGUAGES, AND APPLICATIONS

PRELIMINARY PROGRAM

INVITATION TO OOPSLA '92

John Pugh, Carleton University / The Object People
Conference Chair

On behalf of the Conference Committee, I would like to invite you to participate in OOPSLA '92, the seventh Annual Conference on Object-Oriented Programming Systems, Languages, and Applications.

The conference brings you the very best and latest research, practice, and experience using object-oriented technologies. From a great many submissions, an outstanding set of technical papers, panel discussions, tutorials, workshops, experience reports, and technical demonstrations have been selected.

New additions to the program for '92 are poster sessions and a one-day Educators' Symposium. In addition, the conference will feature an exposition of object technology products and services.

OOPSLA '92 will be held in Vancouver, one of the world's most beautiful and hospitable cities. The spectacular site and unique facilities of the Vancouver Trade and Convention Centre are the perfect setting for OOPSLA. Join us under the "billowing sails".

EXHIBITS

Timlynn Babitsky and Jim Salmons
OOPSLA'92 Exhibits Co-Chairs

Exhibits are an integral part of all OOPSLA conferences. This year's 40,000 square foot exposition of object technology products and services, including a book publishers' fair, is centrally located in the Vancouver Trade and Convention Centre's landmark "five sails" exhibit hall. Exhibits are open concurrently with the tutorial program and the Educators' Symposium on Monday and with the general technical program on Tuesday and Wednesday.

Exhibitors benefit from coffee breaks, lunches, check-in refreshments, and the annual Exhibitors' Reception held in the exhibit hall, and have the opportunity to make scheduled presentations as part of the Vendor Forum. For additional information on exhibits please contact the co-chairs at:

Phone/Fax: +1-803-957-0648

REGISTRATION & INFORMATION

Carole Mann
OOPSLA '92 Registration Chair

An advance program brochure containing detailed tutorial and workshop information, registration form, housing information, and preliminary technical program information will be mailed in July 1992. If you attended the 1990 or 1991 OOPSLA conference or if you are a member of SIGPLAN, you will automatically receive a copy of the advance program brochure. Otherwise, please contact the Chair at:

Gentleware Corporation
2060 Goldwater Court
Maitland, Florida 32751
Phone: +1-407-628-3602
Fax: +1-407-628-3186
Email: mann@eola.cs.ucf.edu

18-22 OCTOBER 1992
VANCOUVER TRADE & CONVENTION CENTRE
VANCOUVER, BRITISH COLUMBIA, CANADA

TECHNICAL PROGRAM INTRODUCTION

Rebecca Wirfs-Brock, Digitalk Inc.
Program Chair

This year's technical program includes three full days of panels, papers, experience reports and invited talks. It includes 31 technical papers presented in ten sessions. Papers were selected from a record number of submissions, and cover key areas of ongoing research and experience in object-oriented technology.

TECHNICAL SESSIONS

Inheritance

Interfaces and Specifications for the Smalltalk-80 Collection Classes
William R. Cook

Monotonic Conflict Resolution Mechanisms for Inheritance
R. Ducournau, M. Habib, M Huchard and M.L. Mugnier

Combination of Inheritance Hierarchies
William Harrison and Harold Ossher

Software Engineering I

Obstacles in Object-Oriented Software Development
Mehmet Aksit and Lodewijk Bergmans

Object-Oriented System Modeling with OMT
Bernd Bruegge, Jim Blythe, Jeffrey Jackson and Jeff Shufelt

Case Study of Object-Oriented Software Development
Dennis de Champeaux, Al Anderson, Marco Dalla Gasperina, Ed Feldhousen, Floyd Fulton, Matt Glei, Colleen Groh, David Houston, Dave Lerman, Charlie Monroe, Rommel Rah, and Dave Shultheis

Implementation

A Comparative Performance Evaluation of Write Barrier Implementations
Antony L. Hosking, J. Eliot Moss, and Darko Stefanovic

Optimizing Method Search with Lookup Caches and Incremental Coloring
André Pascal and Royer Jean-Claude

Object-Oriented Concurrent Reflective Languages Can Be Implemented Efficiently
Hidehiko Masuhara, Satoshi Matsuoka, Takuo Watanabe and Akinori Yonezawa

Invited Talk

Distributed Object Environments: A Foundation for a New Generation of Applications
Alan Snyder, SunSoft Inc.

Experience Papers

Visualizing Objects: Experiences with Analytical Tools for Exploring Human Computer Interaction Concepts
Peete Hansen and Mike Wangler

The Object-Oriented Implementation of a Document Editor
Paul Calder and Mark Linton

ET++ Swaps Manager: Using Object Technology in the Financial Engineering Domain
Thomas Eggenschwiler and Erich Gamma

Preliminary Defect Data from the Iterative Development of a Large C++ Program
James F. Walsh

Languages and Programming

Declarative Programming in a Prototype-Instance System: Object Oriented Programming without Writing Methods
Brad A. Myers, Dario A. Giuse and Brad Vander Zanden

Prototype-Based Languages: From a New Taxonomy to Constructive Proposals and Their Validation
Christophe Dony, Jacques Malenfant and Pierre Cointe

Constraint Patterns as a Basis for Object-Oriented Constraint Programming
Bruce Horn

Experimental Classification Facilities for Smalltalk
Phillip Yelland

Distributed Objects

Distributed Shared Memory with Versioned Objects
Michael J. Feeley and Henry M. Levy

CACL: Efficient Fine-Grained Protection for Objects
Joel Richardson, Peter Schwarz and Luis-Felipe Cabrera

DROL: An Object-Oriented Programming Language for Distributed Real-Time Systems
Kazunori Takashio and Mario Tokoro

Analysis and Design

The Process of Object-Oriented Design
Dennis de Champeaux, Douglas Lea and Penelope Faure

Tunable Formalism in Object-Oriented Systems Analysis: Meeting the Needs of Both Theoreticians and Practitioners
Stephen W. Clyde, David W. Embley and Scott N. Woodfield

Architectures with Pictures
Raymond J. A. Buhr and Ronald S. Casselman

Concurrency

Communications Facilities on Concurrent Objects?
Yutaka Ishikawa

A Formalism for Real-Time Concurrent Object-Oriented Computing
Ichiro Satoh and Mario Tokoro

Concurrency Annotations
Klaus-Peter Löhr

Invited Talk

The End of Objects and The Last Programmer
Grady Booch, Rational

Operating Systems

Lightweight Shared Objects in a 64-Bit Operating System
Jeffrey S. Chase, Henry M. Levy, Edward D. Lazowska and Miche Baker-Harvey

The Muse Reflective Operating System: The Concept and its Implementation
Yasuhiko Yokote

Software Engineering II

Issues in the Design and Documentation of Class Libraries
Gregor Kiczales and John Lamping

Documenting Frameworks using Patterns
Ralph E. Johnson

What Contributes to Successful Object-Oriented Learning?
Chamond Liu, Stephen Goetze and Bill Glynn

EXPERIENCE REPORTS

Experience reports are short presentations of unrefereed reports describing practical experience applying object technology to production quality software development.

Enterprise Infusion of Object technology
John McGehee, Texas Instruments

Objects in the Life-Cycle
Paul Townsend, MPR teltech Ltd.

The WyCash Portfolio Management System
Ward Cunningham, Cunningham & Cunningham, Inc.

Development of Reusable Test Equipment Software Using Smalltalk and C
Alan Dotts, Tektronix

Creating Well Formed Class Inheritance Schemes in C++
Fred Wild, Cadre Technologies Inc.

Constructing a Class Library for Microsoft Windows™
Lon Fisher, Microsoft

AUTOSPEC: Automatic Motor Specification System
Greg Komar, Cap Gemini America

Chembench: Redesign of a Commercial Application Using Object-Oriented Techniques
Glenn Olander, Biosym Technologies, Inc.

POSTERS

Poster Sessions are an alternative forum for viewing results of object-oriented work; with the opportunity for one-on-one interaction with the authors! Poster themes cover the breadth of object technology, from theory to experiences in applications.

PANELS

Reuse: Truth or Fiction
Moderator, Paul McCullough

Is reuse real or merely a desire? Is reuse inherent to object-oriented systems or available via other paths? Is reuse research or practice?

Object-Oriented Languages Based on Strong, Static Typing
Moderator, Steve Litvintchouk, The MITRE Corp.

Ada, Modula3 and Eiffel are object-oriented languages designed for the combined goals of safety and run-time efficiency. Focus on rationale, design philosophy and requirements for each language with specific examples to illustrate trade-offs.

Ensuring Semantic Integrity of Reusable Objects
Moderator, Webb Stacy, CenterLine Software Inc.

Consumers of reusable components need to maintain the semantic integrity of reused components. Several approaches to describe and enforce semantic integrity have been proposed: class invariants, contracts, inspection gauges, assertions, unit and module testing.

From Events to Objects: The Heresy of Event-Orientation in a World of Objects
Moderator, Larry Constantine

Heretics have developed methods that organize behavior and object abstractions around events or transactional sequences. Panelists will discuss similarities and differences in event-oriented methods and compare them with conventional object-oriented approaches.

Object-Oriented Megaprogramming
Moderator, Peter Wegner, Brown University

Megaprogramming is a potential emerging extension of object-oriented technology that extends software component technology by facilitating the design and management of very large abstractions.

The Role of Methods and CASE in Object-Oriented Development
Moderator, Rick DeNatale, IBM

Continuing the line started by last year's panel "Can Structured Methods Be Objectified?" This panel will explore the broader question: Do CASE tools and methodologies hamper object-oriented development?

The Object-Oriented Software Development Process
Moderator, Dennis de Champeaux, HP Labs

Panelists with object-oriented methodology and software process modeling backgrounds will compare the structured paradigm with the object-oriented paradigm in the process dimension.

Educators' Symposium Report
Moderator: Jim Heliotis, Rochester Institute of Technology

A review of issues raised during the one-day Educators' Symposium.

DEMONSTRATIONS

To provide a venue for the very latest work in applying object-oriented technology, live demonstrations of object-oriented systems will be presented during the conference. The presenters are members of the development or implementation team and will provide technical information about their system.

Demonstrations include both commercial and in-house applications, as well as academic and corporate research efforts. In addition to new efforts, "where are they now" demonstrations of projects shown in previous years will be presented.

Proposals are continuing to be accepted. Please submit a one page abstract and technical requirements to:

Rebecca Joos,
OOPSLA '92 Demonstrations Chair
Motorola, Inc.,
6501 William Cannon Drive,
Austin, Texas 78735
Phone: +1-512-891-3617
Fax: +1-512-891-3798
Email: beckyj@oakhill.sps.mot.com

TUTORIALS

Tutorials offer a forum for educating professionals and give attendees the opportunity to take an in-depth look at topics of their choice. This years' program features the widest range of advanced topics ever presented and is taught by some of the leading researchers, teachers, and practitioners in the object-oriented community.

The Pragmatics of Building Object-Oriented Systems
Grady Booch, Rational

The Analysis and Design of Distributed Systems
Mehmet Aksit, University of Twente, Holland

Specification Techniques for Object-Oriented Software
Mahesh H. Dodani, University of Iowa

Object-Oriented Concurrent Programming
Jean-Pierre Briot, Institute Blaise Pascal, France

Types for the Working Programmer
Andrew Black, DEC

Types for the Language Designer
Jens Palsberg &
M. Schwartzbach, Aarhus University, Denmark

Hardware Support for Object-Oriented Systems
Mario Wolczko, University of Manchester, England

Efficient Implementation of Object-Oriented Languages
Craig Chambers , University of Washington
David Ungar, Sun Labs

Constraint-Based Languages and Systems
Bjorn Freeman-Benson, University of Victoria, Canada
Alan Borning, University of Washington

Visual Programming Languages from an Object-Oriented Perspective
Allen L. Ambler, University of Kansas
Margaret Burnett, Michigan Technical University

Writing Efficient C++ Programs
Scott Meyers, Brown University

The Design and Management of C++ Class Libraries
Arthur Riel, Vanguard Training

Intermediate Smalltalk: Practical Design and Implementation
Trygve Reenskaug, TASKON, Norway

Writing Efficient Smalltalk Programs
Ken Auer, Knowledge Systems Corporation

Simulation with DEVS-CLOS
Suleyman Sevinc, University of Sydney, Australia

Advanced CLOS and Meta-object Protocols
Jon White, Lucid

Introduction to Object-Oriented Database Systems
David Maier, Oregon Graduate Institute

Schema Updates for Object-Oriented Database Systems
Roberto Zicari, Johann Wolfgang Goethe Univ., Germany

Teaching "object-think" with Multi-Sensory Engagement
Peter Coad, Object International, Inc

Object Engineering
R. Stonewall Ballard, Component Software Corp.

Object-Oriented User Interfaces
Dave Collins, IBM

Evaluating Reusable Class Libraries
Timothy Korson, Clemson University

Testing Object-Oriented Software
Edward Berard, Berard Software Engineering

Object-Oriented Geometry and Graphics
Krc-Jadiny &
Augustin Mrazik, ArtInAppleS, Czechoslovakia

The Design of an Object-Oriented Operating System: A Case Study of Choices
Roy Campbell &
Nayeem Islam, University of Illinois
Peter Madany, Sun Microsystems

Application Development with the DEC Trettis Object System
Bruce Bullis, DEC

Concepts of Object-Oriented Programming
Raimund Ege, Florida International University

Object-Oriented Project Management
Kenneth Rubin, ParcPlace Systems

Object Design for Modularity, Reuse and Quality
Douglas Bennett, Design Ways

A Comparison of Object-Oriented Analysis and Design Methods
Martin Fowler

A Case Study of Domain Analysis: Health Care
Martin Fowler
Thomas Cairns, St. Mary's Hospital Medical School, England

Integrating Analysis and Design Methods
Derek Coleman &
Patrick Arnold, HP Labs, Bristol, England

Introduction to Object-Oriented Design
Lori Stipp &
Grady Booch, Rational

Teaching Object-Oriented Programming and Design
Jim McKim, Hartford Graduate Centre

EDUCATORS' SYMPOSIUM

For the first time, OOPSLA will offer a special one-day program specifically designed for computer science educators. The Educators' Symposium will provide a forum for members of academic institutions interested in getting started, or sharing experiences, in teaching object-oriented concepts as part of a normal computer science curriculum.

INVITED SPEAKERS

Finding an Educational Perspective for Object-Oriented Development
Linda Northrup, Visiting Professor, US Air Force Academy

On Teaching the Next Generation to Manufacture Software
Wilf LaLonde, Carleton University/The Object People

PAPER PRESENTATIONS

Speakers will present their experiences in presenting object-oriented technologies to students at the undergraduate and graduate level. Themes include object-oriented programming as a first experience in programming, using object-oriented analysis and design as a means of presenting the software engineering discipline, using findings from the field of psychology in the area of learning patterns and enablers, choice of language, and the retraining of experienced programmers.

PANEL DISCUSSION

Speakers from our pool of presenters will participate in a panel discussion at the end of the day. The topic will be "How might the object-oriented paradigm change the way we teach introductory programming, both at a deep conceptual level, and at a fundamental curricular level?"

WORKSHOPS

Workshops provide a forum for experts in an area to discuss current topics and present their latest work. Attendance is through submission and acceptance of a short position paper related to the topic of the workshop. Papers are due August 15, 1992. Refer to the advance program for submission information.

Objects in Large Distributed Applications
Peter Dickman, European Research Consortium In Informatics and Mathematics

Object-Oriented Analysis Processes
Dennis de Champeaux, Hewlett-Packard Labs

Application of Object-Oriented Technology to Document Management
Adrienne J. Kleiboemer, The MITRE Corp.
*Manette Lazear, The MITRE Corp.

Data + Behavior
Bill Harvey, Robert Morris College
*Haim Kilov, Bell Communications Research

Object-Oriented Languages: The Next Generation
Craig Chambers, University of Washington
Fred A. Cummins, EDS
*Mamdouh Ibrahim, EDS
Andreas Paepcke, Hewlett-Packard Labs

Team Approaches to Object-Oriented Design
Steven Fraser, Bell-Northern Research, Canada

Patterns for Object-Oriented Development
Peter Coad, Object International

Applications of Smalltalk in Scientific and Engineering Computation
*Richard Peskin, Rutgers University
Kent Beck, First Class Software

Object-Oriented Metrics
Sam Adams, Knowledge Systems Corporation
Teri Roberts, Object International
*Raj Tewari, Temple University

Development Processes for Use in the Object Paradigm
James O. Coplien, AT&T Bell Laboratories

Towards an Architecture Handbook
Bruce Anderson, University of Essex, England

The Bottom Line: Using Object-Oriented Development in the Commercial Environment
Suzana Hutz, Motorola

An Evaluation of Object-Oriented Technology in Real Time Systems
Christine Anderson, Ada 9X, Elgin AFB
*Mohamed E. Fayad, McDonnell Douglas
Stephen J. Mellor, Project Technology Inc.
Wei-Tek Tsai, Univ. of Minnesota

Analysis and Design — Are Domain Objects the Right Objects?
Magnus Christerson, Objective Systems, Sweden

CLOS
Chris Richardson, Franz Inc.

Experiences with Use Cases and Similar Concepts
Fredrik Lindström, Objective Systems, Sweden

* - contact person

Celesstin: CAD Conversion of Mechanical Drawings

Pascal Vaxivière
Ecole Supérieure des Sciences et Techniques de l'Ingénieur de Nancy
and Centre de Recherche en Informatique de Nancy

Karl Tombre
Institut National de Recherche en Informatique et Automatique
and Centre de Recherche en Informatique de Nancy

A prototype CAD conversion system extracts higher level structures for knowledge-based analysis. It recognizes such entities as screws, ball bearings, and shafts.

Mechanical engineering companies that implement computer-aided design and manufacturing systems must convert their archives of paper drawings to a format suitable for CAD. Digitization of a drawing creates an image of several million black and white pixels that represent the original drawing with varying accuracy, depending (among other things) on the resolution of the scanning device and the quality of the original. However, this raster image information is not directly suitable for CAD systems, which operate with basic structures such as lines and curves. Therefore, many commercial systems[1] have been developed to convert raster images to vector representations suitable for vector editing. These systems may also provide additional modules for text/graphics separation, optical character recognition for the text layer, or semi-automated symbol recognition using simple template-matching techniques. But no commercial system actually provides higher level modules that "understand" a drawing's technical elements. Developing such a system is difficult because of the number of specific CAD/CAM applications (architecture, mechanical engineering, city maps, electrical schemes, and so on) and their diversity of contextual knowledge.

Technical drawings are based on schemes specific to the main activity of the company that uses them. Often, the company's CAD system library stores specifications for screws, bearings, gearboxes, and so on for quick retrieval and insertion into new drawings. The Celesstin system presented in this article is a working prototype that converts a mechanical engineering drawing into a format suitable for CAD (see the sidebar titled "Celesstin prototype"). In developing this system, we stressed recognition of technical entities. Mere vectorizations, such as those offered by existing commercial systems, are too low level to be really useful in a CAD context. Our system is based on the assumption that even when the

0018-9162/92/0700-0046$03.00 © 1992 IEEE

vectorized drawing is distorted, it can be correctly interpreted by using higher level expertise — that is, knowledge about the representation rules used in technical drawings and about the manufacturing technology associated with the represented objects.

Celesstin prototype

Celesstin is a prototype system for converting mechanical engineering drawings into the format of the Dassault Systems' Catia CAD system. Celesstin has been under development for several years at the engineering school of Ecole Supérieure des Sciences et Techniques de l'Ingénieur in Nancy, France. At the end of each academic year in June, a new integrated version of the system is released with the most up-to-date versions of the algorithms designed in the project, completely integrated with a user-friendly interface and the appropriate modules for connecting the program to the host Catia CAD system. Celesstin runs on an IBM 9370-60 machine with a 5080 graphics workstation.

The aim of the Celesstin project is to prove the feasibility of knowledge-based interpretation of drawings. As a prototype, Celesstin has several limitations:

- The input paper drawings must be relatively clean, without lots of "noise."
- Celesstin only processes representations of assemblies such as gearboxes made of rotating axes and of motion transmission and guiding devices.
- Vectorization is limited to straight lines. Celesstin does not recognize circular arcs.
- The system's knowledge rules are strongly based on the drawing's compliance with the French Afnor standard, especially the stipulation that thin and thick lines can be discriminated.

We have tested this prototype only on a limited set of drawings. Hence, it would be pretentious to claim recognition or rejection rates or to propose the use of this system in production. Our goal with Celesstin is to prove that a working production-quality system can be built, not to build one.

Guided tour of Celesstin

Celesstin integrates several modules in a blackboard-based interpretation system, as Figure 1 shows. Once a drawing has been digitized, a first processing step separates the text and the dimensioning lines from the pure graphics part. Celesstin vectorizes the graphics part and assembles the resulting lines into blocks, the basic elements for the technical entities that it creates. The result is transferred to the CAD system. Celesstin tries to match the extracted entities with the corresponding models from the CAD library. It puts the remaining blocks and lines into different layers of the CAD description.

Scanning and preprocessing. Technical drawings are input using a 600-dpi scanner. Then the user can apply image preprocessing tools, the most important of which clean up the bitmap. The user can choose from several sets of cleaning masks — morphological filter operators[2] that eliminate isolated black pixels or small black or white blobs.

Vectorization. The vectorization module converts the image bitmap into a set of lines. For Celesstin we adapted the method proposed by Lin et al.,[3] which consists of splitting the image into meshes and looking at the intersection between the lines and the mesh borders. The result of vectorization is a description of the drawing as chains of vectors having given thicknesses and intersecting at junction points.

Postprocessing. Celesstin analyzes the data structure resulting from vectorization to extract two line-thickness classes: thin lines and thick lines. Celesstin does this automatically by histogram analysis of line thicknesses over the whole drawing. Some line types are very important in technical drawings because they symbolize higher level information such as symmetries, hidden contours, or the presence of matter. Therefore, a procedure searches the vectorization result for specific elements such as dashed and dot-dashed lines and hatching areas (regularly spaced, parallel thin lines across a closed polygon drawn in thick lines).

An interactive tool lets the user correct misinterpreted lines or change line attributes. Figure 2 shows an original raster image and a screen dump of the workstation during the vector correction phase. The vectorization results (Figure 2b) are displayed with different color codes for the different line types: thick lines in white, thin lines in pink, contours of black blobs in red, and dashed and dot-dashed lines recognized as single elements. The user can correct interactively the attributes of the segments (selected line in blue).

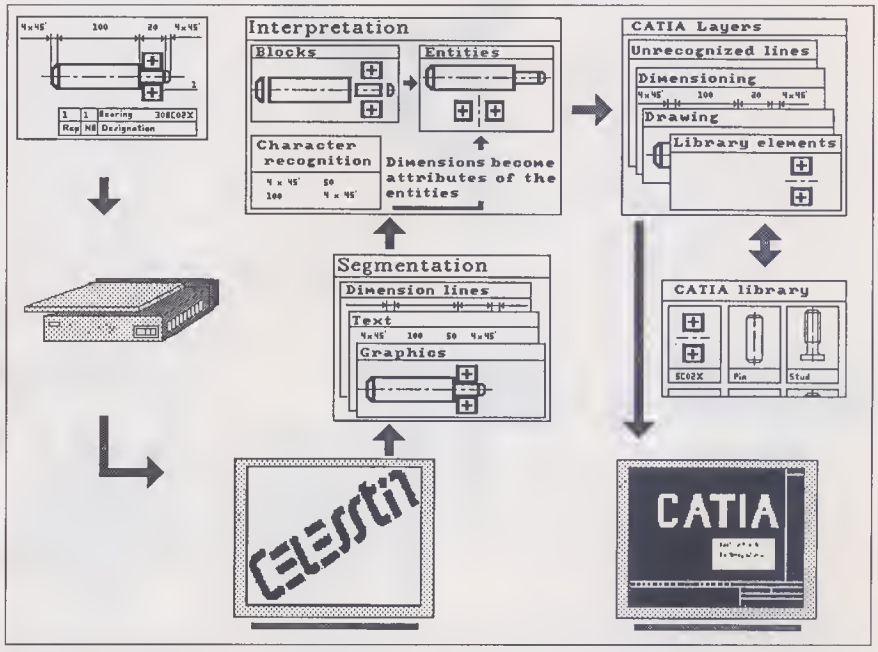


Figure 1. Overview of the Celesstin system.

Vectorization

Vectorization is a very important stage in a technical document interpretation system. It consists of converting a raster image to an image described as chains of vectors and junctions between these chains.

Many vectorization methods have been proposed, but the only approach implemented in commercial systems and thus developed as production-quality code uses skeletonizing, followed by a polygonal approximation of the skeleton.[1,2] A major drawback of this approach is that skeletonizing is based on a view of the image that is too local. This may result in distorted vectors and erroneous choices when the system computes the position of the junction points or looks for circular arcs. Instead of this method, we adapted to mechanical engineering the diagram processing algorithm proposed by Lin et al.[3]

The method is based on a priori knowledge of the maximal thickness of a line and the minimum distance between two lines. Such knowledge is easily available in mechanical engineering, where the following rules apply in France (French standard NF E 04-103):

- The thickness of a thin line must be one fourth that of a thick line.
- The distance between two thick lines must be at least 0.8 mm.
- The distance between two hatching lines varies between 1.5 and 2 mm.

The method consists of splitting the binary image into $n \times n$ meshes, where n is larger than the maximal line thickness and smaller than the minimum distance between two lines. Thus, at most one line in a given direction should cross each mesh, and any line will not overlap more than two meshes. Figure A shows the principle: The vectorization algorithm looks only at the borders of the meshes and approximates the lines by straight segments crossing each mesh.

Lin et al. compiled a library of 51 characteristic meshes that apply in most cases. All the meshes in a given image cannot be coded by an element of this library, so we have an extra "?" code, which corresponds to approximately 5 percent of the meshes in a given image vectorization. We added to the original method a recursive splitting of the meshes into smaller ones, until all the meshes can be given a known code. The mesh is split orthogonally to the first border that contains at least a sequence black-white-black. Celesstin does the splitting in the white interval between the two black intervals. If possible, it splits so that no ambiguity remains on the opposite side of the mesh. Otherwise, it splits arbitrarily. Celesstin applies the splitting operation recursively to the new meshes created, until no unknown sequence remains. For more efficient line following, it keeps only links between split meshes that are overlapped by a common line (see Figure B).

From this point, we further improved the method by going progressively from a "geographic" splitting of the image as a set of regular meshes to a "geometric" splitting where each mesh contains characteristic geometric features. To locate more precisely the angles between two lines, which are very important in technical drawings, we want an angle to be in the middle of a mesh. Such precise localization prevents oblique lines from becoming "stairway-like" during vectorization.

This requirement led us to the concept of *dynamic* meshes, which arrange themselves in the most convenient way for line following. First, by applying offsets to the mesh structure, Celesstin recenters — on a single mesh — lines that overlap two neighboring meshes. In addition, to reduce the total number of meshes used to describe a drawing, Celesstin lets dynamic meshes "grow" by merging two adjacent meshes having the same code into a single mesh. Figure C shows how this reduces the number of characteristic meshes from 51 to 16, corresponding to the following families of cases: junction/intersection, angle, straight line, and black blob.

Finally, from the mesh data structure, Celesstin creates a line structure by following lines from one mesh to the other. During line following, if Celesstin finds two neighboring meshes covered by the same black area, it considers the area as a black blob and codes it by its contour. The result of this vectorization process is a data structure describing the image as segments of different types (thick line, thin line, and contour of black blob) and junctions between these segments.

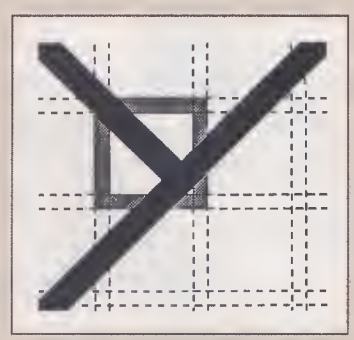


Figure A. Principle of the vectorization method.

References

1. R.W. Smith, "Computer Processing of Line Images: A Survey," *Pattern Recognition*, Vol. 20, No. 1, Jan. 1987, pp. 7-15.
2. V. Nagasamy and N.A. Langrana, "Engineering Drawing Processing and Vectorization System," *Computer Vision, Graphics and Image Processing*, Vol. 49, No. 3, Mar. 1990, pp. 379-397.
3. X. Lin et al., "Efficient Diagram Understanding with Characteristic Pattern Detection," *Computer Vision, Graphics and Image Processing*, Vol. 30, No. 1, Apr. 1985, pp. 84-106.

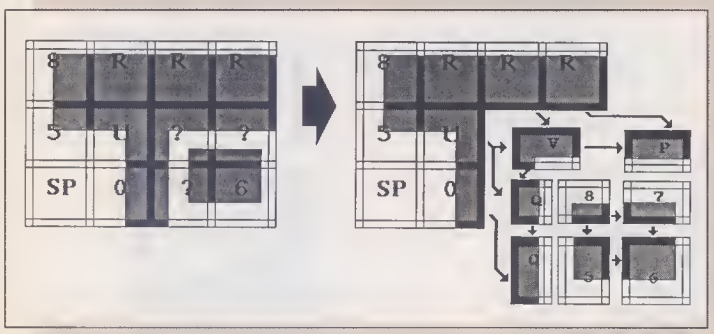


Figure B. Mesh splitting and links between split meshes.

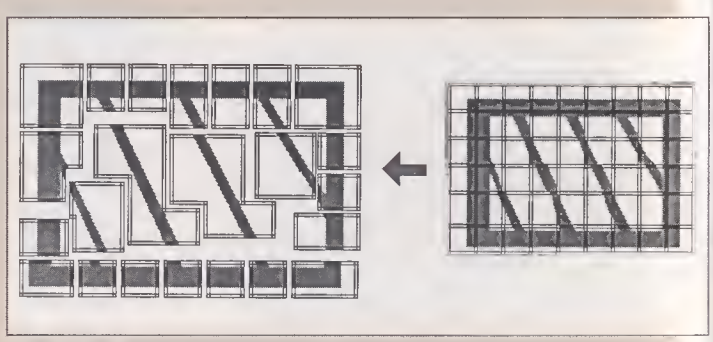


Figure C. Dynamic meshes.

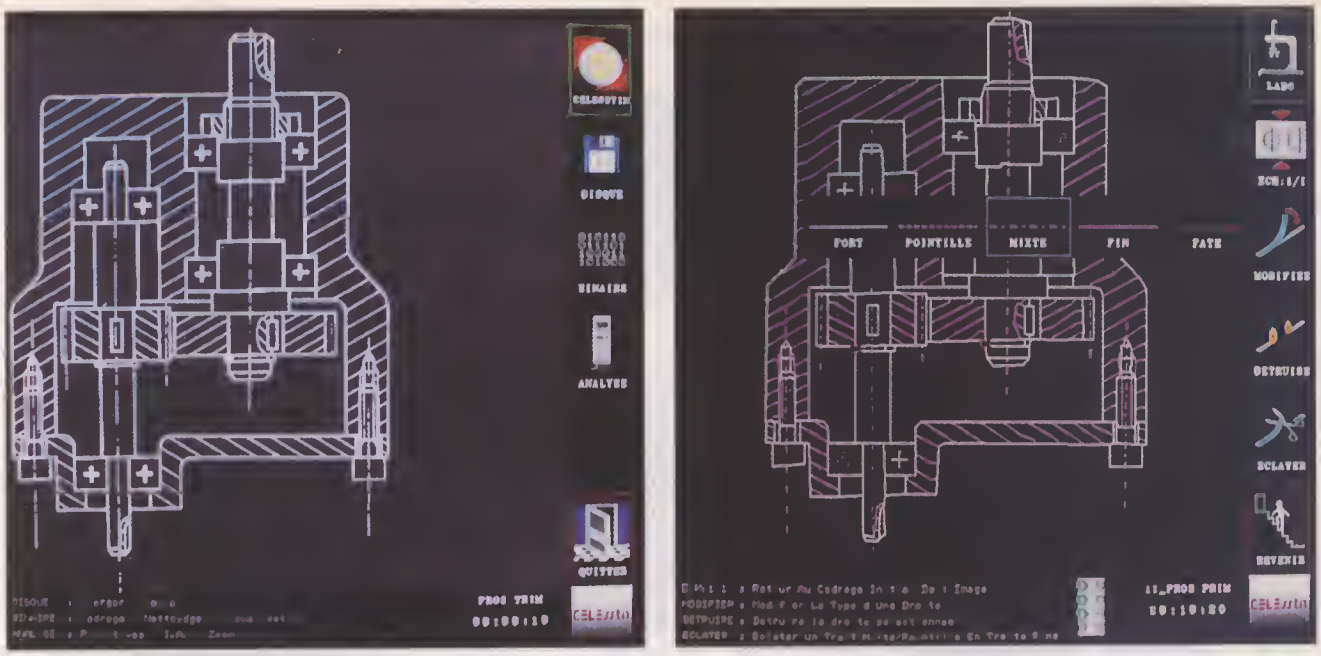


Figure 2. Original raster image (a) and result of vectorization with interactive editing facility (b).

Blocks. Because the structural level of single line vectors is too low to be really useful in the interpretation phase, we decided to create a new description of the vectorized drawing. The basic element of this description is a minimum closed polygon made of thick lines, called a *block*. All the thin lines attached to a block are its *attributes*.

Interpretation. The interpretation module works on the drawing's block description. Its basic strategy is to follow the dot-dashed lines, which are supposed to be axis lines, and to identify technical entities along these axes. The interpretation module continues the analysis by identifying surrounding objects. The module studies the disassembling possibilities and the transmission of motion in the setup to provide high-level knowledge rules for the final stage of interpretation.

Transfer to CAD. Once Celesstin has recognized the drawing's high-level technical elements, it must transfer the information to the CAD system. Most CAD systems compose a drawing as the superposition of several layers. At any time a user can manipulate only a subset of these layers. The highest layer contains the elements recognized as instances of library models. The text layer contains the nomenclature and other annotations, and a dimensioning layer gathers all the dimensions found during analysis. Descending the hierarchy, other layers contain identified blocks that are not part of higher level entities; lines that are not part of any recognized block, themselves classified according to their thickness (thin lines and thick lines); and, in the lowest layer, the residues of segmentation (the small connected components not recognized as being part of graphics or text).

User interface. Celesstin's user interface provides interactive tools and easy access to the whole system. As Figure 2 shows, the hierarchical, icon-based interface lets users choose the tools they want for image cleanup, vectorization, decomposition into blocks, interpretation by the expert system, and export to the CAD system. Most tools combine an interactive part, often reduced to the input of parameters, and an automated part for processing information. Some tools are much more interactive. For example, the vector editor allows manual postprocessing of the vectorization result.

Vectorization and line recognition

Vectorization consists of converting the bitmap into a set of line vectors. For our vectorization, we implemented and improved the method proposed by Lin et al.[3] (see the sidebar on vectorization). After vectorization, Celesstin performs polygonal approximation[4] on the resulting structure to reduce the number of segments. This step also includes a first classification of the line segments into thin and thick lines.

Celesstin corrects the resulting lines by eliminating small irregularities that would disrupt further analysis. Case 1 in Figure 3 shows the correction for contours of black blobs. The black areas must be precisely delineated and represented by the corresponding contour. Case 2 shows the removal of small barbs. If there is no line nearby, Celesstin removes isolated short line segments con-

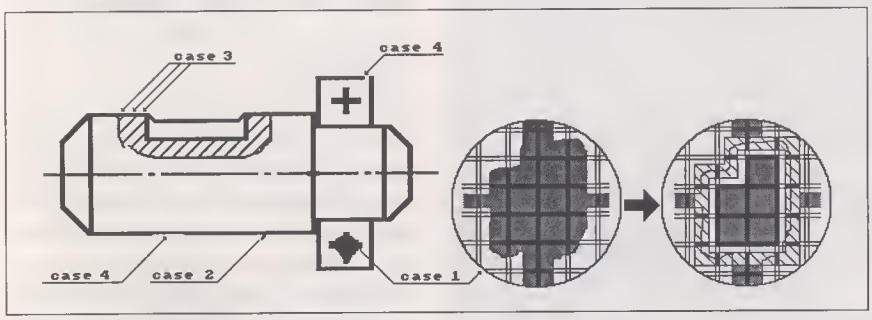


Figure 3. Line structure corrections.

nected to other segments only at one end. In case 3, all couples of aligned segments having the same thickness and meeting at a junction point are merged into a single line. In a technical drawing, a line doesn't arbitrarily change from thick to thin, and case 4 shows the application of continuity rules to correct the line thickness in different configurations.

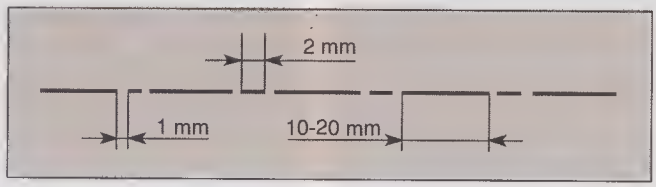


Figure 4. French standard NF E 04-103 for dot-dashed lines.

The next step is to detect dot-dashed lines, representing axis lines, and dashed lines, representing hidden lines. Dot-dashed and dashed lines are very important because they symbolize symmetry lines on which many elements are centered (shafts, screws, and so on). Several methods have been proposed for this purpose; for example, Kasturi et al.[5] use a structural approach based on linking free ends of segments. Our method works in three steps:

(1) location of candidate line segments, that is, segments aligned and at an appropriate distance from one another;
(2) recognition of the line type, which can be dot-dashed (as in Figure 4) or dashed (only short segments); and
(3) replacement of the set of line segments by a single line entity with the attribute dot-dashed or dashed.

We use the standard definition of such lines, as presented in Figure 4. Celesstin classifies candidate line segments into long and short segments and labels the whole line as dashed or dot-dashed, depending on the proportion of short-long-short sequences of segments. As shown in Figure 2b, where a dot-dashed line has been selected (outlined in blue), Celesstin recognizes the line and codes it as a single entity in the final result of line recognition.

In addition, other elements that Celesstin finds during analysis can lead to the hypothesis of a dot-dashed line. Because such lines represent symmetries, Celesstin considers a set of segments to belong to a dot-dashed line if the blocks that this set crosses are symmetrical with respect to it. This allows for better discrimination than the first method in such cases as a dot-dashed line traversing a hatched area.

Of course, the line-recognition phase does not yield perfect results. For instance, the hatching lines in Figure 2 are not all straight, because vectorization is based on local operators that sometimes give a false approximation of the line direction. But Celesstin easily corrects these distortions at the higher levels of interpretation.

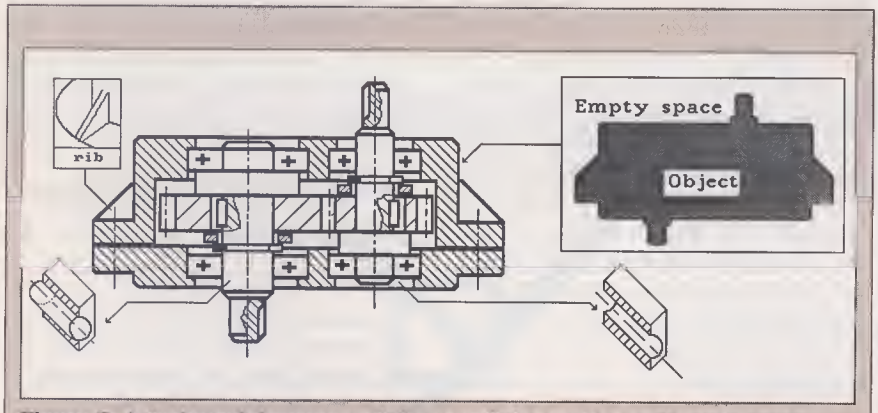


Figure 5. A technical drawing and the search for matter and empty space.

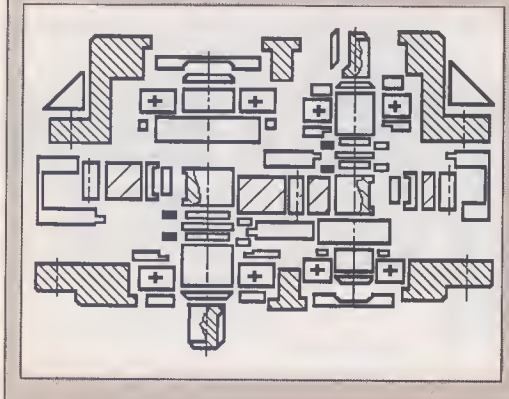


Figure 6. Decomposition of the drawing in Figure 5 into blocks.

Blocks

A principal difficulty in our approach is finding basic elements on which to conduct reasoning. At an earlier stage of this research, we attempted to design techniques for identifying subsets of lines such as hatching and for performing pattern matching between the lines yielded by vectorization and models of CAD objects. We eventually concluded that line-segment features are too low level and do not carry sufficient technical information. We therefore chose to work on a higher level element, the *block*.

In a technical drawing, thick lines represent the orthogonal planar projection of the contours of a mechanical device. Because the thin lines do not outline the device, we decided to ignore them in the first phase. Any minimal closed polygon with thick lines sharing at least one edge with the object's external contour has to enclose matter. If the polygon is hatched, it lies in the section plane; if it is empty, it may represent a piece in front of or behind the section plane (see Figure 5). Any minimal closed polygon with thick lines sharing an edge with one of the polygons previously described will in turn represent a change in geometry (change in diameter of a shaft, for instance), a

new piece in contact with the previous one, or empty space. Simple rules allow Celesstin to formulate hypotheses about these blocks using their thin-line content and that of their neighbors. These rules propagate the labeling of all blocks to the whole drawing. Celesstin adds all thick lines not included in this block decomposition to the content of their including block; they represent keyways, ball bearings, or bores. The result is a complete decomposition of the drawing as a set of structures at a level higher than vectors — that is, at the level of minimal closed polygons, or blocks, drawn with thick lines and their thin-line attributes. Figure 6 shows the decomposition of the drawing in Figure 5.

Celesstin's next operation on the blocks determines their technical attributes. Before this analysis, a block's content is only graphical: It is an empty block or filled area (black block), or hatched with parallel thin lines, which may be parallel to a side of the block. The analysis changes this content into an attribute. The assigned attribute may be "black block" or "white" — in other words, unknown. But the analysis usually assigns an attribute such as matter, represented as hatching, threading, or partial section — or empty space that can be tapped.

During this analysis, a block's content can modify its neighbor's attribute. For instance, Figure 7 shows a section in a tapped element. The dot-dashed line indicates that the hole is cylindrical. Both blocks 1 and 3 contain a threading line, but actually it is block 2 that is tapped. The hatchings in blocks 1 and 3 indicate a section in matter; the tapping lines have been put on these blocks because block 2 represents empty space. In such a case, Celesstin's analysis separates the two families of thin lines and associates the tapping lines with block 2. Finally, the system replaces the "unknown" nature of block 2 with its technical attribute: "tapped" (and hence behind the section plane).

The rules used for this analysis are based on typical rules for engineering drawings (French standards NF E 04-104 and NF E 04-012), as illustrated in Figure 8. When the system finds more than two families of parallel lines, it returns to the previous case by labeling two of the families as hatching.

The result of this analysis is a new, complete description of the drawing, no longer as a set of vectors but as a set of blocks characterized by their technical attributes. Even if Celesstin cannot assign a precise attribute to each block, the description contains "confidence islands" from which the interpretation stage can start to identify technical components such as shafts, screws, and bearings.

Interpretation

We started our work on interpretation by analyzing the way engineers look at drawings that have neither dimensioning nor nomenclature. Engineers generally follow the main dot-dashed

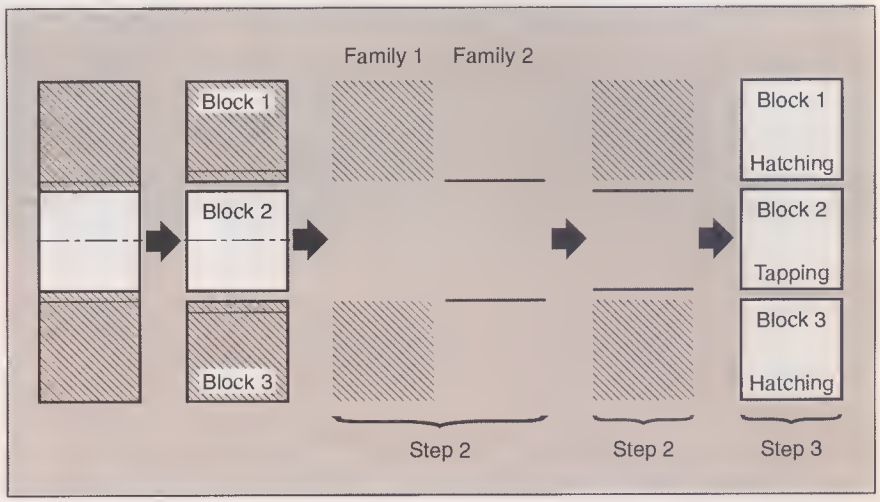


Figure 7. Tapping.

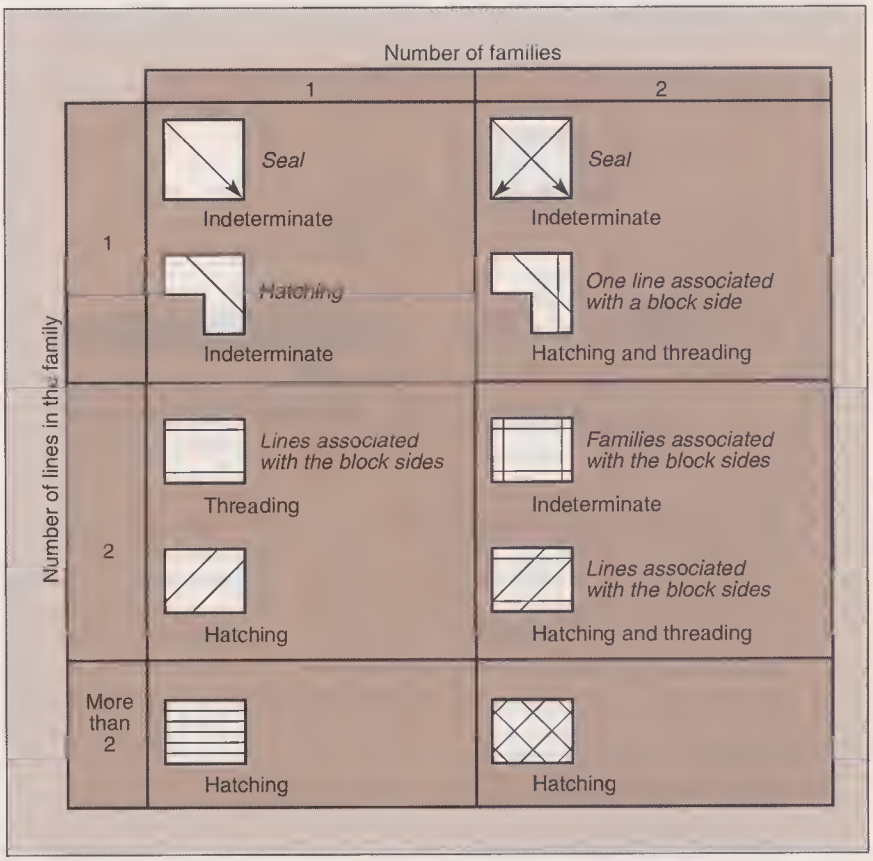


Figure 8. Determining the attributes of blocks.

Recognition of shafts

The knowledge-based interpretation system in Celesstin uses blocks as its basic elements and combines them into more elaborate components to which it can give semantic labels. Here we illustrate the reasoning process and knowledge rules for recognizing shafts. This reasoning works on all blocks located along a dot-dashed line. The basic idea is to start from islands of confidence found by labeling blocks with their technical attributes and to propagate the matter along the axis line.

Some block attributes are *strong indications* of the presence of matter. A good starting point for the recognition of a shaft is

- a threaded block, or
- a block containing a partial section and located on only one dot-dashed line.

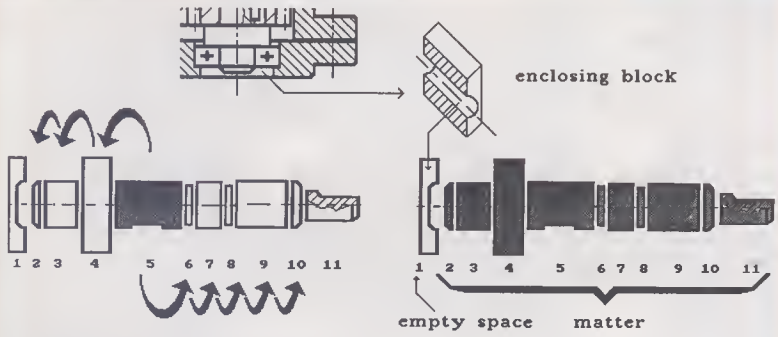
If Celesstin finds no strong indications to initiate the recognition, it can use *weaker indications*, such as the presence of an "enclosing" block representing the background plane of a shaft.

When Celesstin finds a starting point, it applies the following propagation rules:

- Any rectangular or trapezoidal block, symmetrical with respect to the dot-dashed line and neighbor of a block already recognized as matter, is also a matter block.
- Any block containing a partial section (not necessarily rectangular or symmetric) and extending the shaft represents matter.
- Any closing trapezoid (chamfer) on the shaft stops the propagation of matter.
- Any block enclosing a matter block is supposed to be empty, as it usually represents empty space or the casing in the background of the section plane.
- Any tapped block is empty.

The figure below shows the analysis process along the dot-dashed line supporting the gear in Figure 5. Because the whole set of blocks contains at least one partial section, this is a shaft and not a screw. Celesstin thus describes the set of blocks linked in this way with the symbolic notation "shaft."

Such rules are never absolute, but globally such a propagation method remains robust. The strong rules have never failed in our experiments. For recognition failures with weak rules, a last resort is user intervention. In addition, higher level phases of the interpretation may bring into question the proposed labeling and lead Celesstin to propose other hypotheses.



Propagation of matter

Block 5: partial section ⇒ starting point.
Propagation to the left: 4 ⇒ 3 ⇒ 2.
4 and 3 are regular. 2 stops propagation, as it is a closing trapezoid.
Propagation to the right: 6 ⇒ 7 ⇒ 8 ⇒ 9 ⇒ 10.
Same thing happens: propagation stops with 10.

Block 11: partial section ⇒ new starting point.
No propagation; all neighbors are already processed.

Propagation of empty space

Block 1: white "enclosing" rectangle ⇒ empty space.

Propagation of matter along an axis line.

lines, crossing the whole drawing. Then, after they recognize the different components placed on these axis lines, they try to understand the motion-transmission mechanisms. Hence, Celesstin's main interpretation strategy is to focus first on the blocks located on dot-dashed lines, which symbolize a symmetry. The sidebar, "Recognition of shafts," explains how the system applies this strategy.

Celesstin's hypotheses about each technical element must of course be consistent with the remainder of the drawing. Celesstin can analyze a drawing at a higher level by looking at the disassembly and kinematics of the whole setup. But even when Celesstin applies only basic reasoning and simple matching procedures to the blocks it recognizes, it can analyze the drawing in terms of technical elements taken from a CAD library.

However, structuring in terms of blocks is too weak when a system must apply rules such as

- any mechanical device can be disassembled, and
- if a shaft has been recognized, this implies motion and hence a coherent kinematic scheme.

Therefore, we had to devise a new structure, which we call the *entity*. Celesstin creates an entity in two configurations:

- a set of blocks symmetrical with respect to a dot-dashed line and having the same shape, and
- a set of blocks having the same hatching pattern.

After it creates entities, Celesstin can apply higher level knowledge to these new structures. A very strong technological rule is that it *must* be possible to disassemble a mechanical setup. When Celesstin has identified the shafts and screws in the drawing, it creates all possible entities and computes the degree of freedom with respect to one another of all "matter" entities. Celesstin applies a set of knowledge rules using these computational values and tries to progressively disassemble the device. When an entity hinders the disassembling, Celesstin analyzes it more precisely to determine if it represents empty space or a locking device (key, clip).

In addition, a kinematic analysis of the drawing determines the functional role of each entity located around a shaft. The general aim of this analysis is to determine whether an entity touching a

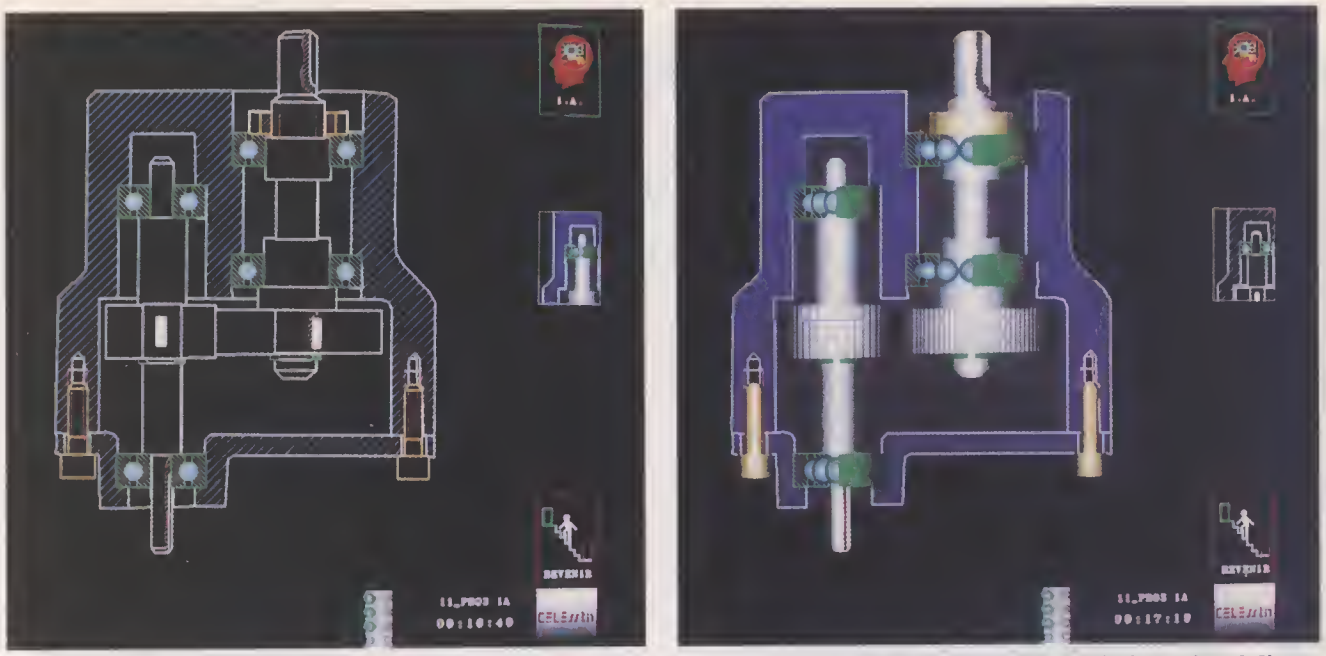


Figure 9. Result of the interpretation process: (a) attributes and recognized entities; (b) two-and-a-half-dimensional display.

shaft rotates with the shaft (for example, the entity could be a gear) or allows the rotation of the shaft with respect to a motionless part such as the casing. This latter configuration identifies bearings and rings.

We implemented the interpretation system in Atome, a blackboard-based multiagent expert system shell.[6] The knowledge rules are relevant only if the drawing represents a usable technical device. Figure 9 shows the result of interpretation of technical entities. In the screen shot in Figure 9a, Celesstin has replaced some entities with the corresponding representation from the CAD library (for instance, the ball bearings) and has associated symbolic attributes such as hatching with some blocks. As this is an interpretation in terms of technical entities, Celesstin can display the recognized drawing in two and a half dimensions (Figure 9b).

Ongoing work and perspectives

The Celesstin system demonstrates the possibilities for formalizing the knowledge mechanical engineers use to read and draw technical documents. We have started studying the different levels of knowledge that can be used in technical document interpretation. As in many other areas, it is possible to distinguish a *structure*, on which a *syntax* can be applied, and *semantics*.[7] For Celesstin, the vectors, blocks, and entities are structural units on which we defined a syntax for combining lower level structures into higher level ones. At the semantics level, we introduced knowledge about mechanical engineering itself — for instance, rules for propagation of matter and for analysis of the drawing in terms of disassembly and motion transmission.

We are adding new modules to Celesstin, including a module that analyzes the structure and syntax of the dimensioning and nomenclature,[8] coupled with a character recognition system for reading the actual numbers of the dimensions and the names in the nomenclature that may be part of the drawing. The character recognition system adds another means of interpretation and transfer to the CAD system: If a drawing annotation refers to an entry in the nomenclature, the system can look up the corresponding entity in the CAD library of models and, by simple pattern matching, check the correspondence between this model and the annotated part of the drawing.

Celesstin is a research product and has some weaknesses. Its representation of knowledge should be better structured, as a "flat" set of rules rapidly becomes hard to manage. Also, its implementation of vectorization is probably less robust and efficient than the vectorization facilities provided by commercial systems. But commercial systems do not provide any high-level analysis of technical entities, which is Celesstin's aim and unique feature. Of course, user interactivity will always be needed to retrieve really usable CAD descriptions of a drawing, but the time and effort necessary for user corrections must be much less than the time and effort needed to create a complete new design in CAD.

Celesstin is in no way a universal, "omniscient" knowledge-based system, knowing everything about any conceivable technical drawing in the world. It is rather a prototype and an integration platform for several research activities (vectorization, knowledge-based interpretation, dimension analysis, and so on). We don't believe that a universal system can be built, as knowledge and aims are very different in different applications. But Celesstin shows a possible path for designing working systems that solve the "transfer from paper to CAD" problem in many specific applications. ■

Acknowledgments

This work was partially funded by a research grant from IBM France to ESSTIN. We thank all the students who worked on the successive versions of Celesstin and all colleagues who have taken part in the work, especially Julian Anigbogu, Yannick Chencvoy, Vincent Chevrier, and Suzanne Collin.

At GTE Laboratories, we have assembled one of the strongest groups of researchers and developers in the world to support GTE's telecommunications businesses. We are the largest local exchange telephone company and the second largest mobile service provider in the U.S. Current opportunities include:

Software Engineers
AIN

GTE is pursuing an aggressive program in Advanced Intelligent Networks with the development of a testbed to prototype new network and operation systems. We have several positions for individuals to design and develop innovative prototype software for GTE's Intelligent Network Testbed.

Candidates must possess an MS or PhD in Computer Science, Electrical Engineering, Mathematics or a related discipline and at least 3 years' working experience in software development. A strong background in software development, methodologies and database design and implementation (ORACLE/Informix), plus experience with C/C++ and UNIX® are required. Experience with X-Windows, Motif, network programming, system administration and telecommunications is preferred. **Box RCS.**

Distributed Computing Researcher

This individual will work with a small group involved in developing distributed computing systems which integrate a large-scale, wide-area, high-speed network of heterogeneous computer systems. This will require conducting original research and proof-of-concept prototype development in the area of distributed operating systems as applied to current and future technologies within the telecommunications industry. A strong background in applied research, software development and software engineering as well as C/C++ and UNIX are required.

Candidates should possess an MS or PhD degree in Computer Science or a related discipline and 5+ years' experience in the design and implementation of operating systems, communications, distributed systems, and/or distributed applications. Experience in telecommunications is highly desirable. **Box REJ.**

We offer an outstanding benefits package including an on-site fitness facility, medical/life/dental insurance, pension, savings and investment plans. Please send your resume to the Employment Department, GTE Laboratories Incorporated, 40 Sylvan Road, Waltham, MA 02254. An equal opportunity employer, M/F/D/V.

GTE Laboratories

THE POWER IS ON

UNIX is a registered trademark of UNIX Systems Laboratories, Inc.

References

1. L.S. Wolfe and J. de Wyze, "An Update on Drawing Conversion," *Computer Aided Design Report*, Vol. 8, No. 6, June 1988, pp. 1-11.
2. J. Serra, "Introduction to Mathematical Morphology," *Computer Vision, Graphics and Image Processing*, Vol. 35, No. 3, Sept. 1986, pp. 283-305.
3. X. Lin et al., "Efficient Diagram Understanding with Characteristic Pattern Detection," *Computer Vision, Graphics and Image Processing*, Vol. 30, No. 1, Apr. 1985, pp. 84-106.
4. T. Pavlidis, *Algorithms for Graphics and Image Processing*, Springer-Verlag, Berlin, 1982.
5. R. Kasturi et al., "A System for Interpretation of Line Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 10, Oct. 1990, pp. 978-992.
6. H. Lâasri and B. Maître, "Flexibility and Efficiency in Blackboard Systems: Studies and Achievements in ATOME," in *Blackboard Architectures and Applications*, V. Jagannathan, R. Dodhiawala, and L. Baum, eds., Academic Press, Boston, 1989, pp. 309-322.
7. K. Tombre and P. Vaxivière, "Structure, Syntax and Semantics in Technical Document Recognition," *Proc. First Int'l Conf. Document Analysis*, Vol. 1, AFCET, Paris, 1991, pp. 61-69.
8. S. Collin and P. Vaxivière, "Recognition and Use of Dimensioning in Digitized Industrial Drawings," *Proc. First Int'l Conf. Document Analysis*, Vol. 1, AFCET, Paris, 1991, pp. 161-169.

Pascal Vaxivière is an associate professor at the Ecole Supérieure des Sciences et Techniques de l'Ingénieur de Nancy (ESSTIN). His research interests are in technical document image interpretation.

Vaxivière received a Diplôme d'Ingénieur from ESSTIN in 1978 and a DEA degree from the University of Nancy in 1990. He is currently preparing a PhD dissertation at CRIN/CNRS on knowledge-based interpretation of engineering drawings.

Karl Tombre is a research scientist at the Institut National de Recherche en Informatique et Automatique (INRIA) working in a research group common to INRIA Lorraine and CRIN/CNRS in Nancy, France. His research interests are computer vision, image analysis, and document image interpretation.

Tombre received a DEA degree from the University of Nancy in 1983 and a PhD in computer science from the Institut National Polytechnique de Lorraine in 1987.

Address questions concerning this article to Tombre at INRIA Lorraine-CRIN/CNRS, Campus Scientifique, BP 239, 54506 Vandoeuvre lès Nancy Cedex, France.

Very Large Data Bases

Announcement of the First Issue

The VLDB Journal

The International Journal on Very Large Data Bases

Aim and Scope

The VLDB Journal is a quarterly published journal dedicated to the publication of scholarly contributions to the advancement of information system architectures, the reflection of technological impacts, and the development of novel applications. It presents significant advances in the design, implementation, and evaluation of information systems.

The First Issue

The first issue will be a special issue on "Federated Databases" to appear on July 1st 1992. It will include the following contributions:

- *An Approach to Recovery Management in a Multidatabase System*
  Y. Breitbart, A. Silberschatz, G.R. Thompson
- *Model Independent Assertions for Integration of Heterogeneous Schemas*
  Y. Dupont, C. Parent, S. Spaccapietra
- *Cooperative Transaction Hierarchies*
  M. Nodine, S. Zdonik
- *Survey on Federated Databases and Systems (I)*
  D. Hsiao

Status of Forthcoming Issues

Forthcoming issues will consist of "special issues" on selected topics of current interest, and "general issues". The second issue is a regular issue and will tentatively include the following papers:

- *A la carte: The Integration of Persistent Stores*
  P. Drew, D. Heimbigner, R. King
- *Buffer Management Based on Return on Composition in Multi-Query Environment*
  P. Yu, D. Cornell
- *Survey on Federated Databases and Systems (II)*
  D. Hsiao
- *Invited Survey on Multidatabase Transaction Management*
  Y. Breitbart, H. Garcia-Molina, A. Silberschatz

The following special issues are committed and are coordinated by invited editors:

- Parallelism in Database Systems (ed. M.J. Carey, University of Wisconsin and P. Valduriez, INRIA)
- Deductive Database Systems Prototypes (ed. K. Ramamohanarao, University of Melbourne)

Other topics considered for special issues include, but are not limited to, the following:

- Optimization in Object-Oriented and Complex Object Database Systems
- Temporal and Spatial Database Systems
- Automatic DB Tuning and Load Balancing
- Active Database Systems
- Database Programming Language Systems
- Multiprocessor Database Systems

While the general issues will be devoted to all system aspects of databases, the special issues should give a good overview over a field by combining several related papers into one issue also including surveys or tutorials. Authors having a significant contribution to a specific subject are encouraged to submit it.

Submittal Information

Please contact one of the Editors-in-Chief for further information and the guidelines for manuscript preparation.

Editors-in-Chief of the VLDB Journal

Fred J. Maryanski (EIC for Americas and Asia)
University of Connecticut
U-86, 352 Mansfield Rd.
Storrs, CT 06269-2086 U.S.A.
Tel.: +1-203-486-2421 Fax: +1-203-486-6379
EMail: fred@provost.vpa.uconn.edu

Hans-J. Schek (EIC for Europe, Africa and Oceania)
Swiss Federal Institute of Technology, Zürich
ETH Zentrum
CH-8092 Zürich, Switzerland
Tel.: +41-1-254-7240 Fax: +41-1-262-3973
EMail: schek@inf.ethz.ch

Subscription form

I would like to subscribe to *The VLDB Journal*. My check for US$24 (US$96 for institutions) for a one year subscription is attached. Please send the journal to the following address (North America delivery is via surface, elsewhere via air mail).

To place a multiyear subscription, multiply the annual rate with the number of years (up to 3 years).

Send to:
The VLDB Journal
The Boxwood Press
183 Ocean View Blvd.
Pacific Grove, CA 93950; USA

Name

Street

City

State/Country Zip/Postal Code

❒ Individual subscriber. I have included a check of US$24 (or multiples thereof).

❒ Institutional subscriber. I have included a check of US$96 (or multiples thereof).

This special section presents six reports on document analysis systems.

A Robust Recognition System for a Drawing Superimposed on a Map

Shigeyoshi Shimotsuji, Osamu Hori, Mieko Asano, Kaoru Suzuki
Toshiba Corporation, Research and Development Center, Kawasaki 210, Japan

Fumihiko Hoshino and Toshiaki Ishii
Tokyo Electric Power Company, Computer and Communication Research Center, Tokyo 104, Japan

The popularization of geographic information systems and drawing management systems is shifting the handling of various drawing data from sheets of paper to computers. However, transferring such data through conventional interactive input systems is extremely expensive. Therefore, it has become more and more desirable for computers to automatically recognize a variety of engineering drawings and maps.

The most crucial issue in computerized drawing recognition is how accurately and efficiently the system detects such drawing components as lines, graphic symbols, and characters from an input image, even when the image quality is poor. Although many techniques for line and character detection have been proposed, they work only for a restricted number of drawing types. This is because the techniques are based on recognizing connected components in an image. They assume that character regions consist of small connected components and that a large component is made up of only symbols and lines. In reality, even noise-free drawings do not satisfy this assumption. Instead, actual drawings contain mostly line breaks, line crossings, and characters touching lines — phenomena that complicate recognition and decrease accuracy.

Recently, Kasturi et al.[1] discussed a technique for processing characters that touch lines. However, their technique cannot be applied successfully in cases where all the characters touch a figure line, such as a straight line or circular

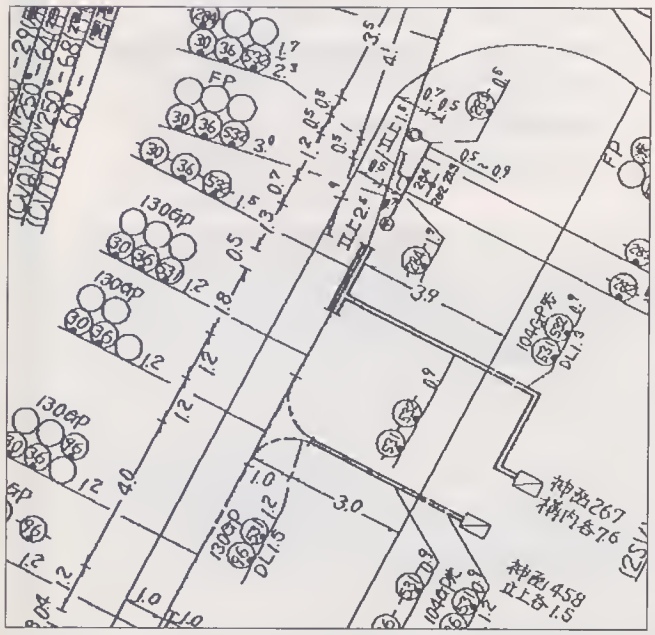


Figure 1. Underground cable diagram for electric power distribution.

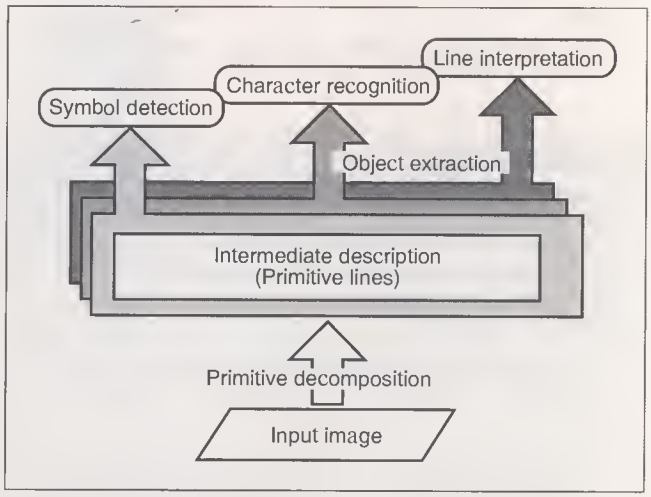


Figure 2. Drawing recognition.

0018-9162/92/0700-0056$03.00 © 1992 IEEE

arc, because their approach to character detection starts from the extraction of small connected components.

We have developed a drawing recognition system that can work efficiently with the line phenomena in real drawings. In our approach, both graphic and character regions of the input image are decomposed into *primitive lines*, the definitions of which include their orientation and connectivity with other lines. Objects are extracted from these primitive lines based on recognition techniques that use shape and topological information to group them into meaningful sets such as character strings, symbols, and figure lines.

We have applied this technique to a recognition system for complicated drawings, such as the equipment diagram superimposed on a map shown in Figure 1. Figure 2 illustrates the information flow in our approach.

Intermediate description: Primitive lines. For efficient and robust recognition of a drawing that includes touching lines and line breaks, our system decomposes lines into primitive segments that are accurately described in terms of both shape and topology. Shape information is represented by a *primary figure* composed of primitive elements. A primary figure here means a straight line, a circular arc, or a filled region. Topological information is described by *connectivity* between primitive elements.

The primitive decomposition process first extracts *skeletons* and *contours* from a drawing image and splits them into short lines by curvilinear approximation. A skeleton refers to a center line for each black region, and a contour is the boundary between black and white regions.

Next, the system analyzes correspondences between skeletons and contours. A region occupied by contour fragments that surround a skeleton segment is defined as a primitive line (see Figure 3). In other words, our system defines a primitive line in terms of a skeleton segment and its corresponding contour fragments. Skeletons are used in describing topological information for the primitive lines. Contours are used in the detection of primary figures.

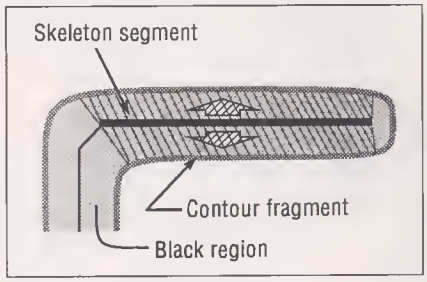


Figure 3. Primitive line.

After the decomposition of an input image into primitive lines, the system detects line breaks by searching for pairs of open endpoints on primitive lines with similar orientations. The connectivity of skeleton segments is used to detect open-ended primitive lines, and contour fragments are used to evaluate their orientation. If a break is found, a new skeleton segment is generated, connecting the terminal points of the break.

This segment enables the detection of straight lines by traversing primitive lines that lie along a straight path. This process employs the position and orientation of the contour fragments of each primitive line. A detected straight line is represented as a set of primitive lines, and a "part-of" relation is attached to each component primitive, indicating its membership in the set composing that straight line.

Circular arcs are detected by grouping primitive lines whose contour fragments form an arc. Detection of a series of adjacent contour fragments employs two measures. One is the k-curvature, which is calculated as the sum of the Freeman codes.[2] The other is a circular approximation made by minimizing the squared error.[3] The first method is fast, but its accuracy is compromised by artifacts introduced during digitization. The second method overcomes this drawback but is much slower. Therefore, we use k-curvature to select a candidate for an arc and refine it by circular approximation. After arc detection, the system merges arcs and tests whether they form a circle.

Drawing component recognition. Recognition techniques based on the intermediate description are used to group the primitive lines into such meaningful sets as symbols, character strings, and lines.

Symbol recognition. This process obtains information regarding the location, type, and orientation of symbols written on the drawing. Our approach uses a model of graphic-symbol shapes both for detecting symbol regions and for identifying symbols.

Most graphic symbols in diagrams can be broken into closed loops, terminal points, and filled regions. Figures with these features can be defined as primitive elements that make up a graphic symbol. The model describing a symbol consists of two parts. One part describes the primitive elements composing the symbol; the other defines the geometric relations between these primitive elements.

We use a hypothesize-and-test method to recognize symbols. When part of a symbol is detected, the system then hypothesizes the whole symbol boundary and verifies it by consulting the model.

Character recognition. This process obtains pertinent information regarding the location, direction, and content of a character string from a drawing. Character recognition is accomplished by detecting a character-string region from an input image and then identifying each character in the string. One difficulty is that character strings are not always written horizontally. Some strings are slanted. Most such strings, however, are written parallel or perpendicular to a neighboring line. Therefore, character-string detection is carried out, first, by grouping short primitive lines that are near each other into character-string candidates. Next, their orientation is examined by searching for a nearby line. If it is found, the orientation of the line is assumed to be the character-string orientation, and the process tests whether the candidate character regions form a string in that orientation. If the test for every neighborhood line orientation fails, the set of primi-

tive lines is rejected as a string candidate.

The character identification phase applies character segmentation to the detected string region and recognizes each character. This process involves a recursive-segmentation-and-recognition method. The system extracts one character region by searching for a break position in a string region and trying to recognize the character pattern. If no pattern is recognized, the system selects another break position for character segmentation and tries again to recognize a character pattern. This process is repeated until the segment reveals a character.

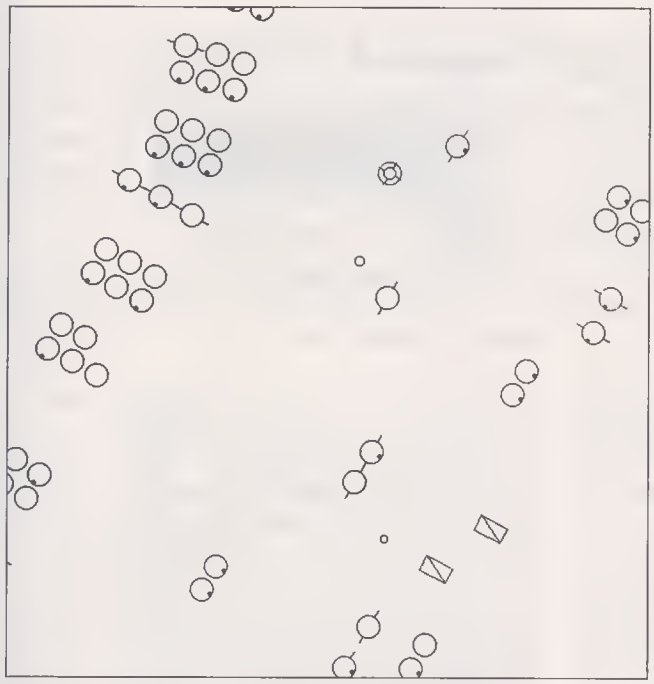


Figure 4. Symbol recognition result for Figure 1.

Line interpretation. Each line in a drawing represents a meaning (for example, a road, a cable, or a comment). For computers to handle lines in a drawing, they must not only distinguish styles, such as solid or dashed, but also analyze meanings. While style is identified by analyzing breaks along the line, meaning cannot be interpreted merely by local feature analysis.

A line style does not correspond uniquely to a meaning. Line meanings are determined by relations to other drawing components. Some lines are interpreted by connected symbols and some by character strings. Some line meanings are determined by geometric relations to other lines whose meanings have already been established. Therefore, interpretation processing requires a method that propagates interpretation from recognized symbols and characters to uninterpreted lines.

Another difficulty in line interpretation is that symbol and character recognition are not complete. There are typically some errors in the recognition of actual drawings. If an iterative interpretation begins at a "misrecognized" symbol, the error propagates to a wide area. Therefore, it is difficult to interpret each line in a deterministic way.

Our approach introduces a probability into the interpretation. Each probability regarding possible interpretations for the meaning of a line is successively updated by a relaxation method that lets the system decide on meaning from global information. The interpretation process attaches probabilities for possible meanings to each line. The initial probabilities are calculated based on connected symbols and characters. Next, these probabilities are updated based on the

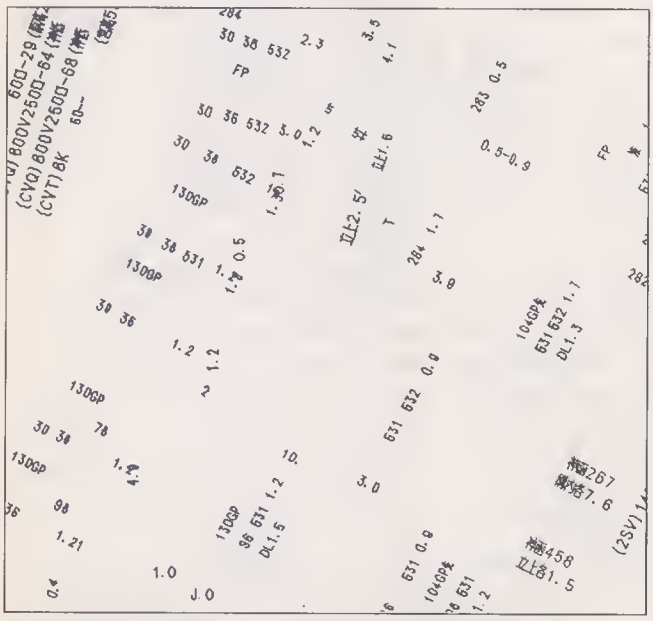


Figure 5. Character recognition result for Figure 1.

Figure 6. Line interpretation result for Figure 1.

Table 1. Recognition accuracy (percentage).

Test Drawings	No. 1	No. 2	No. 3	No. 4
Symbol recognition	98.9	97.6	94.6	98.7
Character recognition	68.4	74.5	59.4	86.9
	(91.4)*	(94.3)	(88.3)	(97.2)
Line interpretation	87.9	93.4	90.7	94.1

*A value in parentheses shows recognition accuracy only for alphanumeric characters.

geometric relations to adjacent lines. After adequate iteration, the meaning corresponding to the highest probability is selected for each line.

Conclusion. The robust and efficient drawing recognition system presented here is based on a representation technique that uses accurate shape and topological line information for an input drawing image. This representation supports primitive decomposition and object extraction that enable accurate automatic interpretation even for low-quality drawings. The proposed system, which is adapted to an underground electric cable diagram, has been implemented on a workstation. Figures 4, 5, and 6 show example recognition results.

We carried out experiments on four A0-size drawings to evaluate the system. (Figure 1 is representative of the complexity of each test drawing.) Table 1 presents the recognition accuracy. Experimental results on actual drawings have shown that the proposed system is effective in real-world applications such as data entry for an equipment management system, although there is still room for refinement of each drawing element recognition technique. ■

References

1. R. Kasturi et al., "A System for Interpretation of Line Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 10, 1990, pp. 978-991.
2. F.C.A. Groan and P.W. Verbeek, "Freeman-Code Probabilities of Object Boundary Quantized Contours," *Computer Vision, Graphics, and Image Processing*, Vol. 7, No. 4, 1978, pp. 391-402.
3. M. Brady and H. Asada, "Smoothed Local Symmetries and Their Implementation," *Robotics Research*, Vol. 3, No. 3, 1984, pp. 36-60.

Shigeyoshi Shimotsuji is a research scientist at the Information Systems Laboratory, Toshiba Research and Development Center. He has worked for Toshiba in drawing recognition, geographic information systems, and image processing hardware since 1983. From 1991 to 1993, he is working as a research affiliate of the Media Laboratory at MIT, Cambridge, Mass. His research interests are document image processing, pattern recognition, and cognitive science.

Osamu Hori is a researcher at the Information Systems Laboratory, Toshiba Research and Development Center, where he has worked since 1986 in image processing. Other research interests include human-machine interfaces that use pattern recognition techniques.

Mieko Asano joined the Toshiba Research and Development Center in 1988, where she has since been engaged in image recognition research and its applications. Her interests include pattern recognition and image processing.

Kaoru Suzuki is a researcher in three-dimensional image processing at Kansai Research Laboratory. His research interests are robotics, computer vision, and optical character recognition.

Fumihiko Hoshino has been with the Computer and Communication Research Center of Tokyo Electric Power Company since 1987, where he has worked in research and development for advanced computer-aided information systems and a recognition system for drawings. His research interests are pattern recognition, object databases, and computer-supported cooperative work.

Toshiaki Ishii has been with the Computer and Communication Research Center of Tokyo Electric Power Company since 1991. He works in the research and development of a recognition system for drawings. His research interests include pattern recognition and human-machine interfaces using pattern recognition techniques.

Reading Handwritten Digits: A ZIP Code Recognition System

Ofer Matan, Henry S. Baird, Jane Bromley, Christopher J.C. Burges, John S. Denker, Lawrence D. Jackel, Yann Le Cun, Edwin P.D. Pednault, William D. Satterfield, Charles E. Stenard, and Timothy J. Thompson*
AT&T Bell Laboratories, Holmdel, NJ 07733

Reading is a difficult task. For speed and accuracy, humans still outperform machines at most vision tasks. Our goal is to design machines that can read handwriting at levels approaching or exceeding human performance. We have developed algorithms, based on neural networks, to read strings of handwritten digits. These algorithms have been incorporated into a system that reads handwritten ZIP codes appearing on real US mail. The techniques we used are applicable to other handwriting recognition tasks.

System overview. The novelty of this system is the recognition-based segmenter. The segmenter is a hybrid of connected-components analysis (CCA), vertical cuts, and a neural network recognizer. Connected components that are

*Matan's current address is Department of Computer Science, Stanford University, Stanford, CA 94305.

single digits will be handled by CCA. CCs that are combined or dissected digits will be handled by the vertical-cut segmenter.

There are four main stages of processing:

- *Preprocessing:* Noise is removed and the digits are de-slanted.
- *CCA segmentation and recognition:* Each CC is evaluated by the recognition unit. If the score is above a certain threshold, the CC is considered solved. Segments with scores below the threshold are handled by the next level of segmentation.
- *Vertical-cut-point estimation and segmentation:* Parts of the image that are not yet considered solved are segmented using vertical cuts. Each segment is classified and scored by the classifier. The segmentation that results in the best combined score is chosen.
- *Directory lookup:* Answers that do not appear in the ZIP code directory are rejected and the vertical-cut segmenter is recalled to generate the next-highest-scoring candidate.

The system was trained and tested on approximately 10,000 images, five- and nine-digit ZIP code fields taken from real mail. The images are black and white, scanned at 212 pixels per inch. The ZIP code database was created by contractors to the US Postal Service. The ZIP code fields were located by humans and extracted by drawing rectangular boxes around the field.

The digit recognizer. Both segmentation components call the digit recognizer, a neural network trained using back propagation. Le Cun et al.[1] described this recognizer in detail. The main feature of this neural network is that its input is a pixel image rather than manually designed feature vectors. The only preprocessing done is normalizing the input image to a 20-by-20-pixel gray-scale image using a linear transformation. The gray levels are scaled to fall within the range −1 to +1.

The output consists of 10 units, one for each digit. The desired output for a pattern belonging to class i is +1 for the ith output unit and −1 for the other output units. This output vector is converted to an estimate of the probability of correct classification by applying the Softmax normalization scheme for classifiers with N mutually exclusive outcomes. This enables us to combine the individual digits' scores into a combined score for a given segmentation.

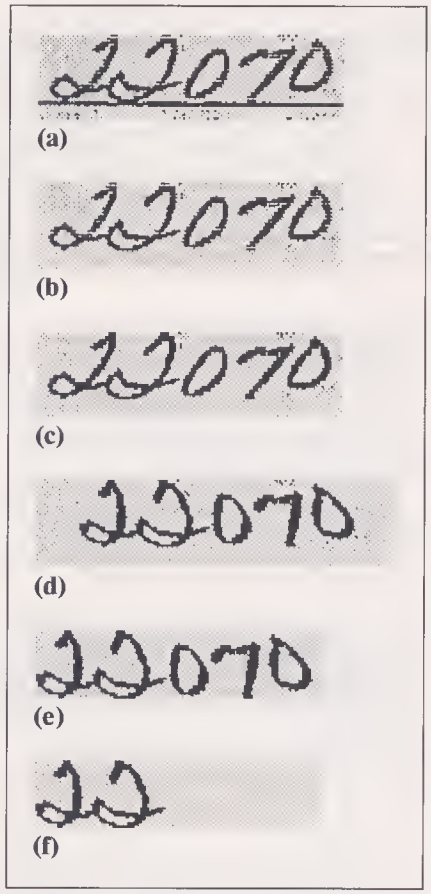


Figure 1. Preprocessing stages and connected-components analysis: (a) original image; (b) delete underline; (c) fill missing scan lines; (d) remove skew and slant; (e) clip border and remove flyspecks; (f) remove digits solved by connected-components analysis.

The network used for the ZIP code recognizer was bootstrapped from a network trained on about 10,000 isolated digits, described by Le Cun et al.[1] After we segmented a considerable number of ZIP codes using the whole system, the network was retrained on the segmented digits, thus tuning it to recognize isolated digits generated by the segmenter it must cooperate with.

Image preprocessing. The preprocessing stage is essential to ensure that the system is robust. The goal is to transform the input image into a standard form with minimum noise. Figures 1b-1e show the preprocessing stages for the ZIP code shown in Figure 1a.

Connected-components analysis. The first task is to identify the connected components. The definition of "connected" has been generalized to include pixels that are very near but not actually touching. Each CC is checked to determine if it is too small or too large to be a reasonable digit. If it is very small, it is called a "flyspeck" and discarded. If it is too large, it probably consists of two or more connected digits and must be handled by the next level of segmentation. CCs that pass the validity check are evaluated by the recognizer. Those that score above a certain threshold are considered solved; the rest are passed on to the next level. Figure 1f shows the parts of our example image that were not solved by CCA. The estimated number of digits determined at preprocessing is reexamined in light of the CCs found.

When all CCs are recognized with scores above threshold and the number of segments is five or nine (or nine and a hyphen), the segmentation is concluded.

Vertical-cut-point estimation and segmentation. At this stage the system must recognize all parts of the image not solved by CCA. Figure 2 demonstrates the vertical-cut segmentation process. The CCs marked as solved are erased from the image and the vertical pixel projection of the remaining image is calculated by passing it through an exponential filter. This gives a value close to +1 for areas of white space and close to zero when the projection value is well over the estimated stroke width. This value gives us the "projection score" of cuts. The resulting function is smoothed and the maxima points are located. These will be the candidate cut-points. A candidate cut-point with a projection score

greater than a certain threshold is termed an obvious cut.

We must find a certain number of cuts, determined by the number of digits that should be in the image (five or nine), minus the number already found. If the number of obvious cuts is equal to the number of necessary cuts, then the set of obvious cuts is the solution. If more cuts are needed, the system uses the recognizer to find them. If there are more obvious cuts than needed, it is assumed that a problem exists with the image, and all obvious cuts will be discarded. The system will then evaluate all candidate cuts using the recognizer.

The cuts determined by the CCA and the obvious cuts found by projection divide the image into a number of segments that we call clumps. Using an estimated pitch, we can get an upper and lower estimate on the number of digits in each clump. In the example shown in Figure 2, there are three clumps, so two additional cuts are needed. The first clump is estimated to have two or three digits, the second to have one or two digits, and the third to have only one. Candidate cuts are generated for each clump so that they have good projection scores and are nearly equidistant.

Each candidate segmentation is sent to the recognizer, and the resulting probabilities are multiplied* to give a score for that clump. The system evaluates all combinations of subsegmentations of the clumps, provided they give the correct number of digits. The best score determines the final segmentation.

The overall number of calls to the digit recognizer is small (nine in our example). For five-digit ZIP codes, the average number of calls is 10. This is a factor of two over the minimum that would be needed even if we knew the correct segmentation a priori.

We have added a small amount of semantic processing, using the fact that not all five-digit numbers are valid ZIP codes. When an invalid ZIP code results, the segmenter will generate the next-highest-ranking interpretation until a valid ZIP code is found.

Segmentation and recognition performance results. Our system was tested on 2,585 images of five- and nine-

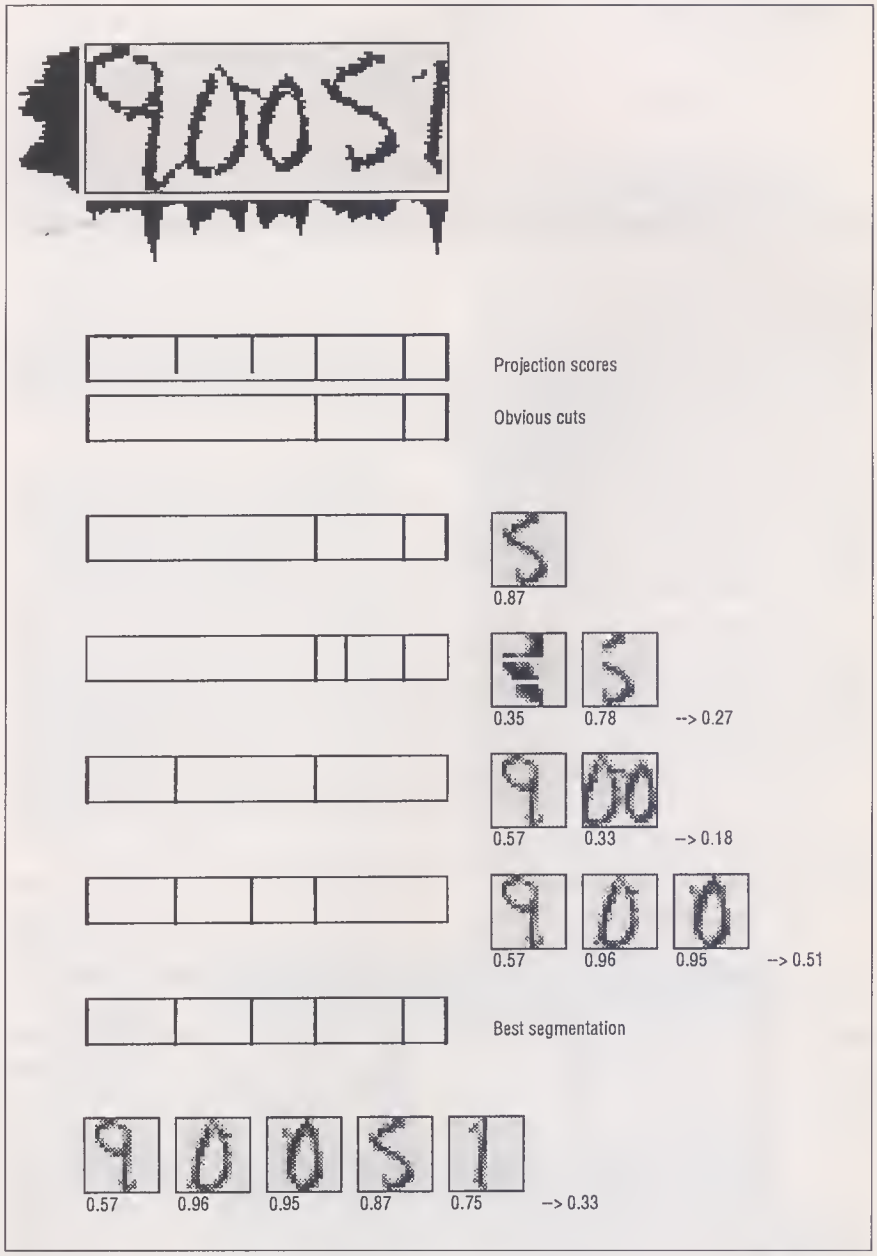


Figure 2. Vertical-cut segmentation. To demonstrate this process, the CCA segmenter has not been applied to this image. The preprocessed image with its horizontal and vertical projections is shown on top. Candidate cuts are represented by lines of a length proportional to their score. Four have high scores; several others are hard to see at the figure s scale. Two scores are so high that they are called obvious cuts, leaving two more to be found. Each row of the figure shows trial subsegmentations of the two leftmost clumps. The images passed to the recognizer and the resulting scores are shown at the right. The best combined segmentation is given at the bottom.

*We make the imperfect assumption that the probabilities are independent and therefore multiplying the individual scores gives an estimated probability for correctness of the digit string.

Table 1. Percentage of raw errors and errors at 40 percent rejection for various versions of the system.

System Version	Raw Error (%)	Error at 40% Rejection (%)
Whole ZIP code	26.15	5.47
First five digits	24.53	4.64
First five digits plus ZIP directory	23.83	3.22
First five digits plus ZIP directory, no CCA	35.59	6.95

digit ZIP codes. Table 1 summarizes the results for various tests.

The basic system achieved a raw error rate of 26.15 percent. When 40 percent of the lowest scoring images are rejected, the error rate for the remaining 60 percent is 5.47 percent. These results are per ZIP code, not per digit, and were obtained without using the ZIP code validation directory.

For the postal application, the crucial digits are the first five. When the measure of correctness depends only on the first five digits, the raw error rate decreases to 24.53 percent. At 40 percent rejection, the error rate is 4.64 percent.

When the ZIP code directory is used to discard illegal answers, the error rate decreases to 23.83 percent raw and 3.22 percent at 40 percent rejection.

Connected-components analysis solves about 35 percent of the cases. We tested the importance of CCA by temporarily deactivating it; the error rate increased to 35.59 percent raw and 6.95 percent at 40 percent rejection.

Implementation. The system was initially prototyped in Lisp using a neural network simulator running on Sun-4 workstations. The system was then ported to C++, also running on Sun-4s. In addition, the code has been ported, with minor changes, to run on an AT&T DSP32C board plugged into an AT&T PC 386 computer.

The average processing time per ZIP code image is approximately 3 seconds and is about equally divided among preprocessing, segmentation, and recognition. Higher speeds could be obtained by optimizing the code. Much higher speeds (tens of ZIP codes per second) should be achievable using custom VLSI chips.

Conclusion. Our system recognizes strings of freehand-written digits at state-of-the-art performance levels. (Examples of other work can be found in the *Proceedings of the Fourth United States Postal Service Advanced Technology Conference*.) The system is based on what we term "recognition-based segmentation." This differs from backtracking in the segmentation recognition process[2] in that a whole set of segmentation hypotheses is scored by the recognizer. The interpretation chosen is the one that gives the best overall score.

This scheme is not necessarily computationally expensive. If the segmentation candidates are chosen carefully, the additional computation is small. We have chosen a hybrid solution using connected-components analysis and pixel projections to generate the candidate segmentations. The average number of calls to the recognizer in the system is only twice the number of digits in the image. The recognizer is a neural network that has been trained to evaluate the candidates generated by the segmenter.

Recognizing freehand-written digit strings is tremendously important. This task is much harder than recognizing machine-printed digits or individual digits written in boxes, but we have shown that it can be done. The basic principles we described should be readily transferable to more general problems, although each new application will require incorporating specialized knowledge of the task. Our next goal is machine recognition of alphanumeric handwriting.

The system described here is our first-generation system. A second generation now under development relies less on heuristics to generate the cuts and is more strongly coupled to the recognizer's training process. Preliminary results achieved by combining our first- and second-generation systems have shown considerable improvement. The raw error rate (per ZIP code) is now well under 20 percent, and the error rate at 40 percent rejection is less than 1 percent.[3] ■

Acknowledgments

Support of this work by the Technology Resource Department of the US Postal Service under Task Order 104230-90-C-2456 is gratefully acknowledged.

References

1. Y. Le Cun et al., "Handwritten ZIP Code Recognition with Multilayer Networks," *Proc. 10th Int'l Conf. Pattern Recognition*, Vol. 2, IEEE CS Press, Los Alamitos, Calif., Order No. 2063, 1990, pp. 35-40.

2. E. Lecolinet and J.V. Moreau, "A New System for Automatic Segmentation and Recognition of Unconstrained Handwritten ZIP Codes," *Proc. Sixth Scandinavian Conf. Image Analysis*, Vol. 1, M. Pietikainen and J. Roning, eds., Pattern Recognition Society of Finland, Oulu, Finland, 1989, pp. 585-592.

3. C.J.C. Burges et al., "Shortest Path Segmentation: A Method for Training a Neural Network to Recognize Character Strings," to be published in *Proc. Int'l Joint Conf. Neural Networks*, IEEE, New York, Oct. 1992.

Ofer Matan is a PhD student in the Department of Computer Science at Stanford University. He was previously a member of the technical staff at AT&T Bell Laboratories. His research interests include machine learning and the relationship between connectionist and symbolic computation.

Henry S. Baird is a member of the technical staff in the Computing Science Research

Center, AT&T Bell Laboratories. His research focuses on algorithms for machine vision, with emphasis on the interpretation of images of printed documents.

Jane Bromley joined AT&T Bell Laboratories in 1990 and has been working on the application of artificial neural networks to human perception tasks, including the reading of handwritten text.

Christopher J.C. Burges joined AT&T Bell Laboratories in 1986 and worked on network transmission performance and signaling-network routing algorithms. Currently he is researching the application of neural network technology to alphanumeric image segmentation and recognition.

John S. Denker is on the technical staff at AT&T Bell Laboratories, where his research interests include the foundations of learning from examples, real-world applications of neural networks, and the unification of neural and classical techniques.

Lawrence D. Jackel is head of the Adaptive Systems Research Department at AT&T Bell Laboratories, which investigates machine learning and machine perception. His current research focuses on neural-net algorithms, hardware, and applications.

Yann Le Cun has been a member of the technical staff at AT&T Bell Laboratories since 1988. His current interests include neural networks and connectionist models, learning theory, pattern recognition, and VLSI implementations of massively parallel systems.

Edwin P.D. Pednault is on the technical staff in the Machine Perception Research Department at AT&T Bell Laboratories. His research includes on-line handwriting recognition, pen-based computing, and shared-work environments.

William D. Satterfield is a member of the technical staff in the Advanced Information Systems Development Group at AT&T Bell Laboratories. He is currently working on image processing and contextual analysis issues pertaining to optical character recognition.

Charles E. Stenard supervises the Advanced Information Systems Engineering Group at AT&T Bell Laboratories. He is AT&T's program manager for applied research and system integration of delivery address interpretation programs for the US Postal Service.

Timothy J. Thompson is a member of the technical staff at AT&T Bell Laboratories. His most recent development activities include image display and optical character recognition applications involving form processing and facsimile.

From Paper to Office Document Standard Representation

Andreas Dengel, Rainer Bleisinger, Rainer Hoch, Frank Fein, and Frank Hönes
German Research Center for Artificial Intelligence (DFKI), PO Box 2080, D-6750 Kaiserslautern, Germany

The task of a paper-computer interface is to transform printed information into a symbolic representation. For integration of such interfaces into office information systems, we must consider standards for electronic document representation into which the information can be transformed. The international standard ODA[1] (Office Document Architecture) provides mechanisms to both generate and exchange electronic documents, plus achieve a fundamental common understanding of their structure.

This article presents the principles of the model-based document analysis system we call ΠODA (Paper interface to ODA) that we developed as a prototype for the analysis of single-sided business letters in German.[2] ΠODA is based on the architecture model of ODA, which has been enhanced to meet the requirements of document image analysis.

Initially, ΠODA extracts a part-of hierarchy of nested layout objects (such as text-blocks, lines, and words) based on their presentation on the page. Subsequently, in a step called *logical labeling*, the layout objects and their compositions are geometrically analyzed to identify corresponding logical objects that can be related to a human perceptible meaning (for example, sender, recipient, and date in a letter). Finally, a context-sensitive text recognition for logical objects is applied using logical vocabularies and syntactic knowledge. As a result, ΠODA produces a document representation that conforms to ODA.

Figure 1 shows all analysis steps and the ΠODA document architecture model. The individual steps are primarily sequential, but allow backtracking. Thus, each step of ΠODA can confirm or reject the results of preordered steps.

Document architecture model in ΠODA. Paper documents usually have

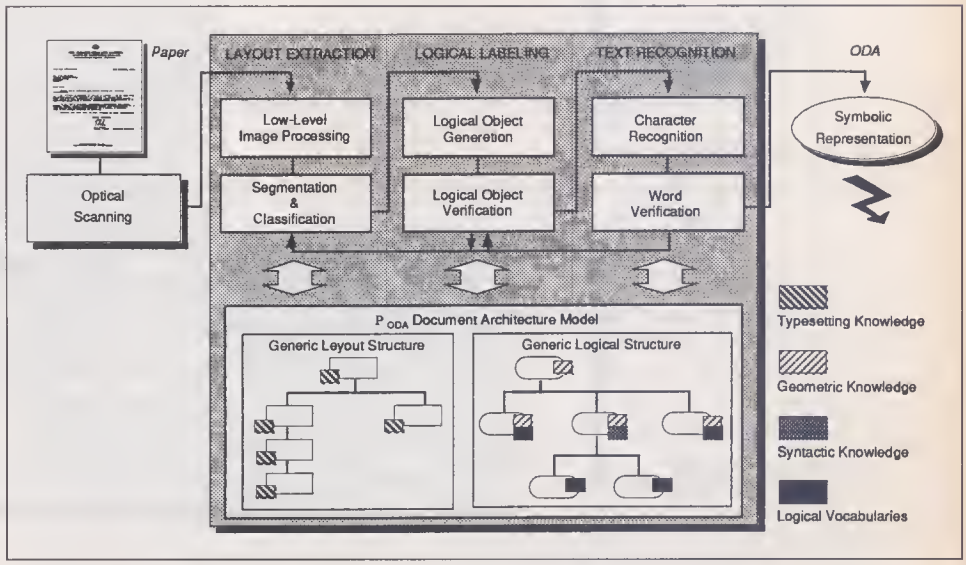


Figure 1. The ΠODA system: Analysis steps and architecture model.

a high degree of structure, a fact that a human reader seldom reflects on but nevertheless uses directly to filter relevant information out of printed data. Certain document classes (for example, scientific papers, letters, or forms) have characteristic structures that can be specifically described.

The ΠODA document architecture includes two generic tree structures, a layout structure and a logical structure. They provide different but complementary views of the same contents. Elements of these structures are called objects. An object that is not subdivided into smaller objects (that is, a leaf of a tree) is called a basic object; other objects are called composite objects. They are considered prototypes (classes) for specific objects (instances) specified by a set of characteristics — a pattern that is common to its members. Such a pattern includes methods for generating specific objects, methods to determine the values of attributes, and methods to control consistency among objects. This architectural basis of ΠODA is analogous to the generic structure concept of ODA.

In ΠODA, the document type "business letter" is represented by a generic logical structure and a generic layout structure that specify and aggregate relevant objects. These structures provide a means to generate an ODA-conforming representation of a given document (that is, its specific layout structure as well as its specific logical structure). Then, the specific basic objects are related to so-called content portions (that is, type text, line drawing, or photograph).

In document image analysis, the objective is to transform document structures from a nonelectronic medium, like paper, into an electronic medium. Here, questions arise regarding how to extract layout objects automatically from a given document image, how to identify corresponding logical objects, and how to recognize the contents of the document. Thus, document image analysis can be considered automatic generation of an electronic document in the sense that it electronically reproduces an existing nonelectronic document. Therefore, the generic structure concept of ODA is an ideal orientation, but one that must be enhanced according to the requirements of document image analysis. For that purpose, knowledge sources are attached to each generic class to facilitate the analysis. For example, typesetting knowledge is associated with layout classes. Geometric knowledge, logical vocabularies, as well as syntactic knowledge (grammar) are associated with logical classes. This enhanced architecture of ΠODA allows a model-based analysis of letter images.

Document image analysis. Transformation of printed business letters into an ODA-conforming format involves three ΠODA analysis steps: layout extraction, logical labeling, and text recognition.

Layout extraction. To obtain a paper-document bitmap representation, we use a 300 dots-per-inch flatbed scanner. For layout extraction, ΠODA initially performs low-level image processing, including image encoding and skew adjustment.

ΠODA uses a segmentation method called *smearing* derived from the run-length smoothing algorithm (RLSA). Smearing is an adequate technique for segmenting down to word level and is easy to implement. However, the method is sensitive to mixed font sizes.

Smearing extracts image blocks rep-

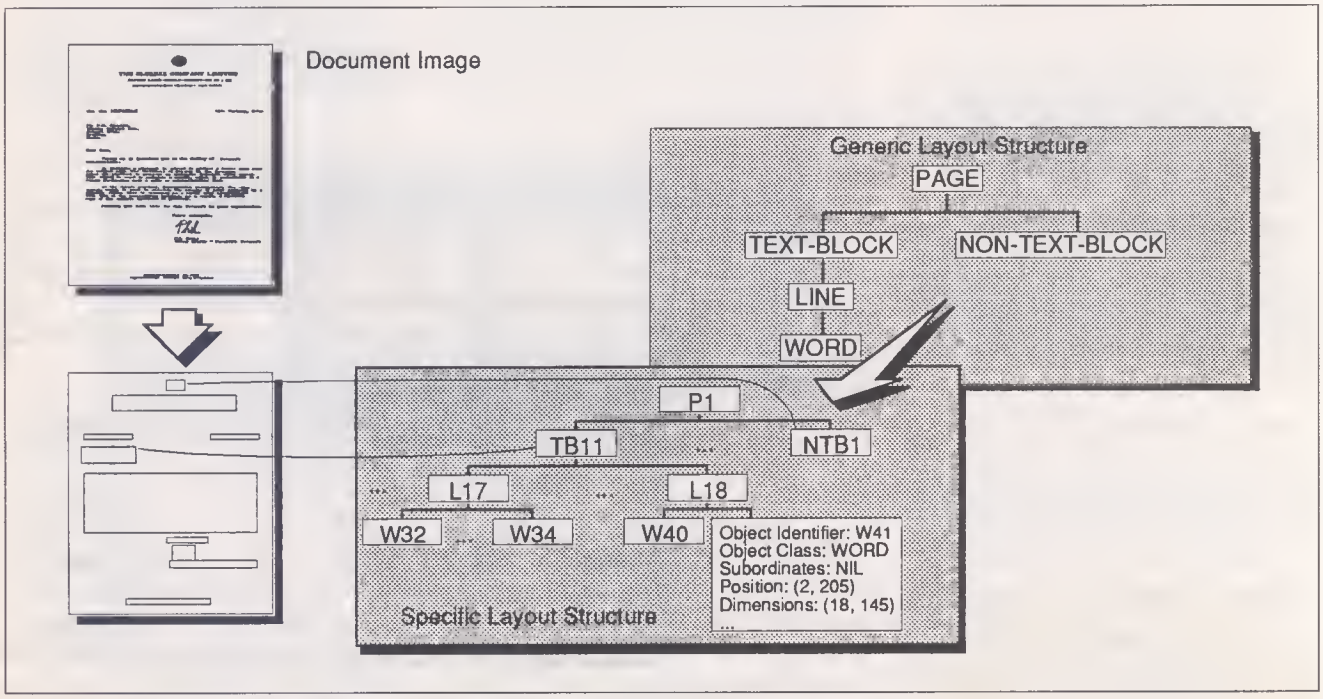


Figure 2. A document image and its corresponding specific layout structure.

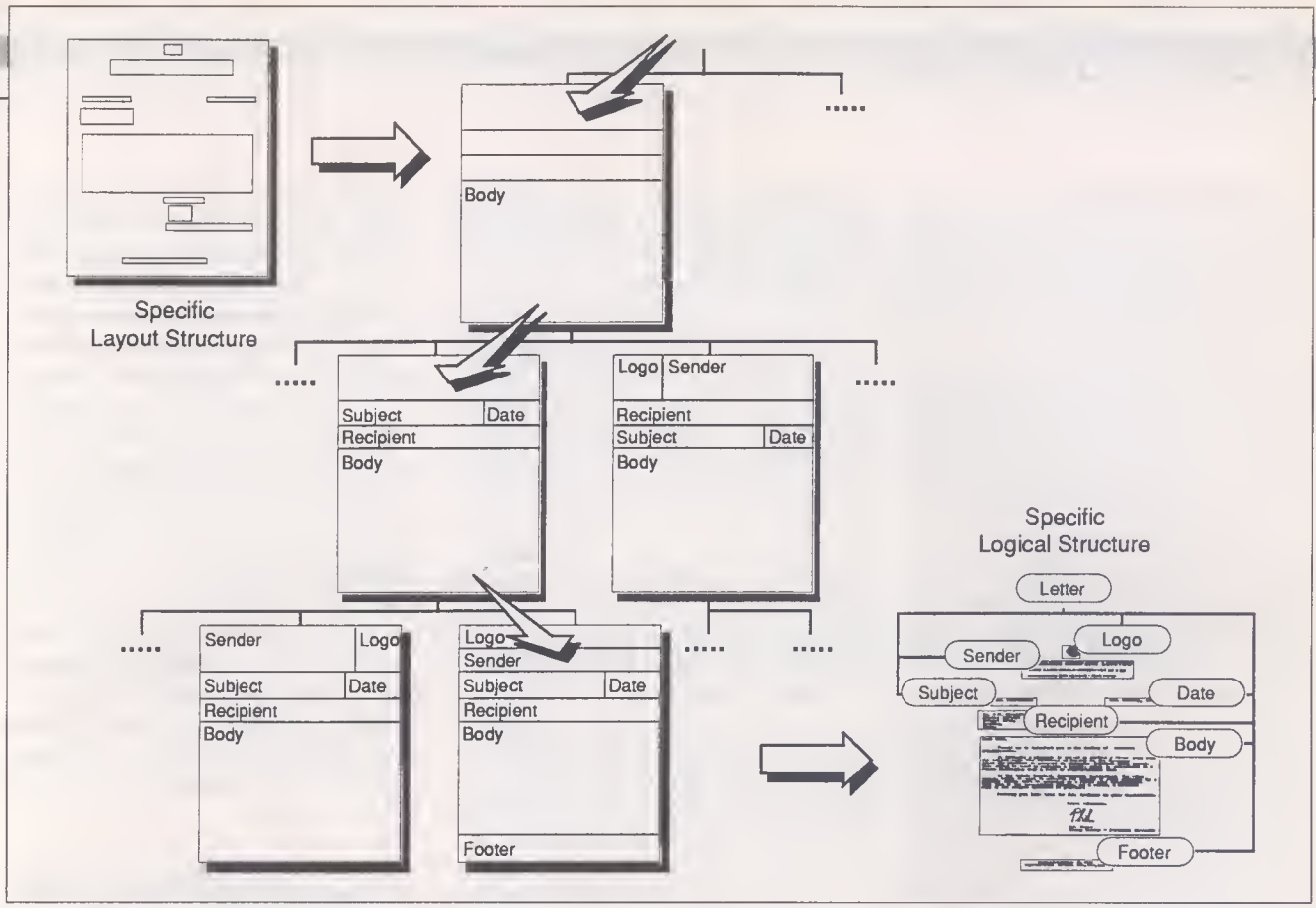


Figure 3. The geometric tree and principles of logical labeling.

resented by specific layout objects. Each object is specified by its position and dimensions, which serve as references to corresponding images. Starting with the raster image of the full page, the segmentation process recursively traverses the generic layout structure, instantiating layout objects and segmenting them into subordinates by applying the associated typesetting knowledge in the corresponding classes. Each traverse stops when a basic object is reached. Thus, the structures inside a TEXT-BLOCK — LINEs and WORDs in our model — are reconstructed by means of specific layout objects (see Figure 2).

Table 1. Some geometric rules for ΠODA recipients.

Feature	Rule	Measures of Belief (MB)	Measures of Disbelief (MD)
vertical_origin	$0 \leq$ vertical_origin ≤ 0.25	0.89	—
	$0.25 <$ vertical_origin ≤ 0.33	0.10	—
	$0.33 <$ vertical_origin	—	0.99
	:		
number_of_lines	number_of_lines < 4	—	0.95
	$4 \leq$ number_of_lines ≤ 6	0.92	—
	number_of_lines $= 7$	0.02	—
	number_of_lines > 7	—	0.99

Logical labeling. Using a hypothesize and test strategy, ΠODA analyzes the specific layout objects geometrically. For that reason, arrangement alternatives as well as local geometric features of logical objects are attached to the generic logical classes as appropriate knowledge sources.

In a *geometric tree*, various arrangements of logical objects within business letters are described on different specification levels[3] (see Figure 3). The root of the tree represents the most general arrangement. Every single-sided document belongs to this class. The internal representation is organized so that logical object arrangements of parental nodes are inherited by children. For our experiments, we use a tree with about 40 terminal nodes.

In addition, an individual logical object can be described independent of the text it contains solely by its shape. Therefore, each generic logical object includes geometric knowledge capturing statistical results obtained from evaluating geometrical aspects of about 190 business letters. In particular, we examined all logical objects relative to their intrinsic geometric characteristics. The probabilities are represented in subsets of rules as measures of belief (MB), and some of them are applied as measures of disbelief (MD = 1 – MB). Table 1 reflects geometric rules for recipients in ΠODA.

In the table, MB and MD values range between 0 and 1. The values in the rules are relative to the width and height of a page. Actually, in ΠODA, 12 logical objects are each described by 11 different geometric features.

Logical labeling of a given document amounts to finding a path from the root of the geometric tree to one of its leaves, thereby matching the specific layout structure stepwise against the alternative arrangements. All nodes visited are instantiated before validating their relevance, so that their evidence can be tagged on. If the refinement in a tree node fits with the specific layout (very little cut shifting is allowed), the appropriate label is assumed to be a hypothesis. In such a case, a specific logical object is generated as an instance of the corresponding generic logical class. For verification, all layout objects associated with the logical object are compared to the geometric rules in the appropriate knowledge slot of its class. The MBs and MDs that are obtained are combined by Dempster-Shafer's rules of combination. All hypotheses are pushed into an agenda, which defines the next step in the tree. If logical labeling fails, ΠODA initiates backtracking to the layout extraction step for resegmentation. Figure 3 shows the representation of the geometric tree indicating the principles of logical labeling.

Text recognition. Here, specific basic layout objects describing word images are sent to a commercial text-recognition system. The output of that system comprises complete or fragmentary strings representing word hypotheses. Because logical labeling provides the identification of document constituents (that is, a restriction of context), the subsequent text verification can be performed in an expectation-driven manner. Consequently, word hypotheses are verified by matching corresponding strings against legal words of specific logical objects (such as a set of names as possible recipients of business letters). In particular, a letter tree (*trie*) is combined with a selective-access matrix that allows efficient matching of fragmentary input strings against legal words.[2]

For refinement of the specific logical structure and controlled selection of vocabularies, syntactic knowledge is attached to some of the generic logical classes — in particular, those representing well-structured parts of a letter (such as address and date). This knowledge determines the order in which subordinated logical objects (for example, ZIP code, city, and street) can occur and, therefore, controls the access to corresponding logical vocabularies. Vocabularies and grammars are used not only to assist the recognition process, but also to confirm or refute hypotheses of logical labeling; therefore, they can lead to a backtracking. As a result, all content portions (see the section entitled "Document architecture model in ΠODA") of type "image" related to layout objects of the word level are converted into content portions of type "text" where alternative and fragmentary results are allowed.

Figure 4 shows an example of the output produced by ΠODA. This representation of a document conforms to ODA.

Conclusions. The ΠODA system has been implemented on Sun Sparcstations. Besides transforming printed information into the international standard ODA, the design of ΠODA reveals a fundamental advantage: All stages of document image analysis, layout extraction, logical labeling, and text recogni-

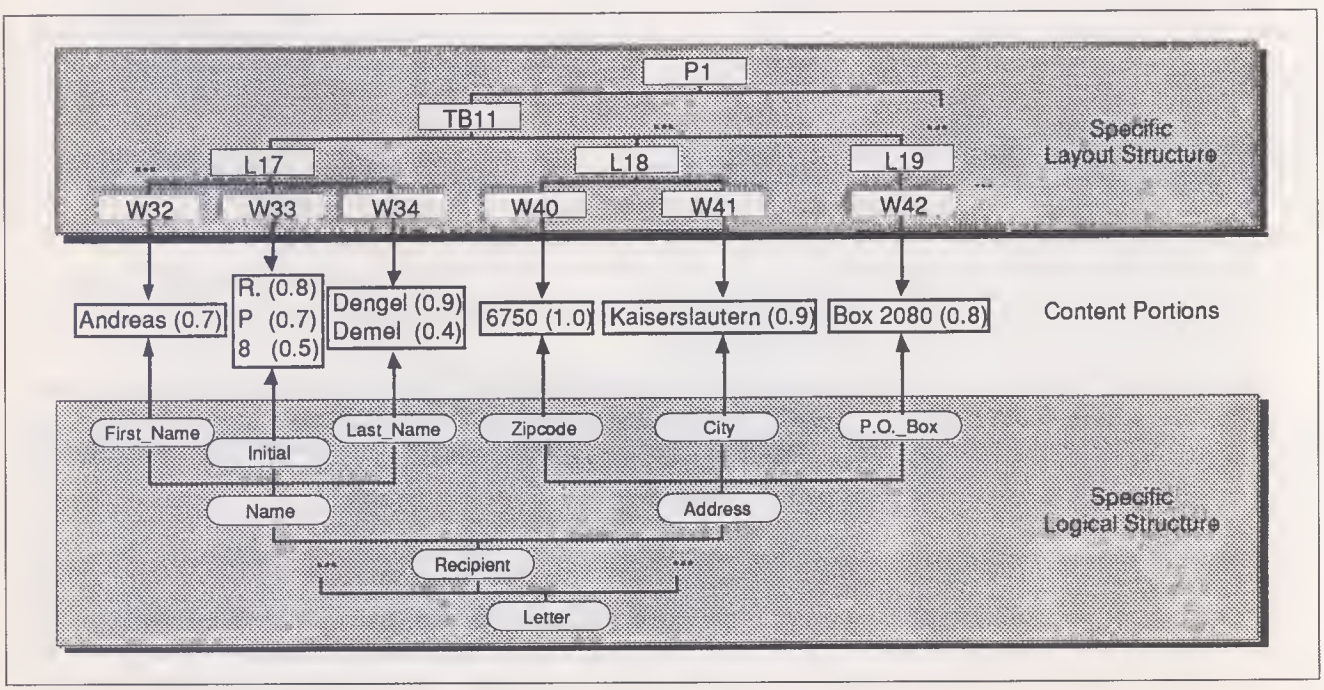


Figure 4. The representation of ΠODA results.

tion are guided by the ΠODA architecture model and additional knowledge. Obviously, such a model-based approach provides optimum flexibility because alternative analysis methods can be used for individual steps where only I/O interfaces of the distinct phases are predefined.

We are now working to eliminate existing weaknesses. One is the lack of a robust and multipurpose technique for text/graphic separation. The commercial system we use produces only one hypothesis for each character. Therefore, we are developing our own character recognition system for obtaining alternative candidates.

Our future work will also concentrate on an expectation-driven partial text analysis of logical objects, including the design of a structured lexicon. ■

Acknowledgment

This work has been supported by Grant No. FKZ-ITW-9003-0 from the German Federal Ministry for Research and Technology.

References

1. W. Horak, "Office Document Architecture and Office Document Interchange Formats: Current Status of International Standardization," *Computer*, Vol. 18, No. 10, Oct. 1985, pp. 50-60.

2. A. Dengel et al., "ΠODA: The Paper Interface to ODA," German Artificial Intelligence Research Center Research Report RR-92-02, Kaiserslautern, Germany, Feb. 1992.

3. A. Dengel and G. Barth, "Anastasil: Hybrid Knowledge-Based System for Document Image Analysis," *Proc. IJCAI 89, Int'l Joint Conf. Artificial Intelligence*, Vol. 2, Morgan Kaufmann, San Mateo, Calif., 1989, pp. 1,249-1,254.

Andreas Dengel has been with the German Research Center for Artificial Intelligence (DFKI) since 1989. He first served as assistant to the director and is currently the head of the Office Automation Research Department. His major research interests are in applications of AI and integration of methods of pattern recognition, linguistics, and psychology.

Rainer Bleisinger has been with DFKI since 1989 and serves as a research scientist. Involved in document analysis, he concentrates on modeling aspects and methods for information extraction. His major research interests are AI techniques, especially temporal reasoning, knowledge representation, and text analysis.

Rainer Hoch joined DFKI in 1991 and serves as a research scientist. His major research concerns the design of a dictionary for document analysis, document representation, hypertext, and partial text analysis strategies.

Frank Fein has been with DFKI since 1990 and serves as a research scientist. His major research interests include image processing techniques and mechanisms for analysis control.

Frank Hönes joined DFKI in 1989 and serves as a research scientist. His research interests are digital image analysis, text recognition, and AI techniques for document analysis.

DRS: A Workstation-Based Document Recognition System for Text Entry

Tomio Amano, Akio Yamashita, Nobuyasu Itoh, Yoshinao Kobayashi, Shin Katoh, Kazuharu Toyokawa, and Hiroyasu Takahashi
IBM Research, Tokyo Research Laboratory, 5-19, Sanbancho, Chiyoda-ku, Tokyo 102, Japan

As hypermedia and multimedia document systems become popular, not only text data but also image data (such as pictures in a document), page layout, and reading order can be stored and used. However, when documents are in the form of printed papers, entering such information requires time-consuming manual operations.

To aid operators in this task, researchers have studied experimental document analysis systems that include optical character recognition (OCR).[1] A practical system on inexpensive hardware would attract many business and engineering users. For this reason, we have developed a workstation-based prototype called the Document Recognition System (DRS). It provides functions for image capture, block segmentation, page structure analysis, and character recognition with contextual postprocessing, as well as a user interface for error correction. All the functions except image capture and character recognition have been implemented by means of software for the Japanese edition of OS/2.

Figure 1 shows the system configuration. The PS/55 workstation (using an 80386 microprocessor with a 25-MHz clock) is compatible with the PS/2 except for its screen resolution and key-

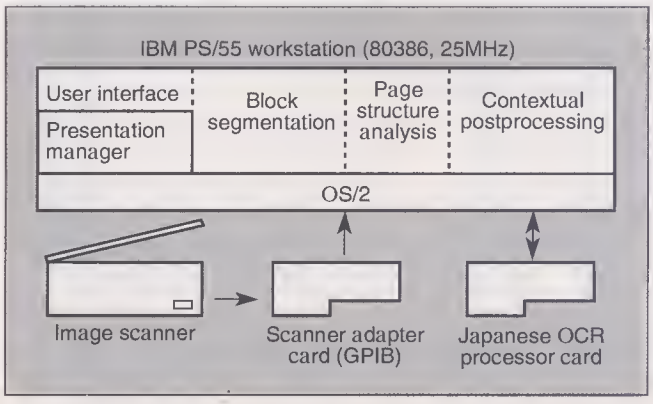


Figure 1. The Document Recognition System configuration.

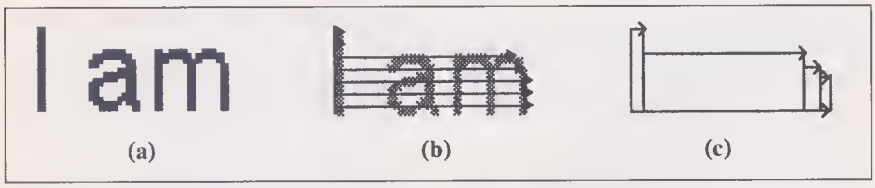


Figure 2. The DRS block segmentation algorithm: (a) original image; (b) smeared run-length data; (c) boundaries and rectangles.

board. A binary page image is captured in 400-dpi resolution by an image scanner that can be shared by several systems. The Japanese OCR processor card is dedicated to recognizing about 3,000 Japanese characters, including Kanji, Kana, and alphanumerics. This card consists of a 68020 microprocessor (with a 16.5-MHz clock), memory for an image and recognition dictionary (templates), and two LSI chips for feature extraction and discrimination. Program codes for the 68020 and a recognition dictionary are loaded from the PS/55 when the DRS is initiated.

Block segmentation. A captured image is first segmented into blocks (rectangles on the image).[1] Most block segmentation methods are based on one of two approaches: segmenting an image into blocks according to projection profiles, or detecting small components by component labeling and then integrating them into blocks. Although both approaches require a large amount of workstation CPU power, a smearing preprocess can reduce processing time in the second approach.

Figure 2 shows the method we use as a substitute for component labeling for a smeared image.[2] During raster scanning, smeared run-length data (starting positions and lengths) are generated by replacing short horizontal white runs with black runs. The top and bottom boundaries are detected by comparing the smeared run-length data from vertically successive lines. A run located under a white run is considered to be part of a top boundary. Bottom boundaries can be detected in a corresponding manner. The coordinates of rectangles sandwiched between top and bottom boundaries are calculated and stored in a buffer.

Several rectangles may be detected for a single character string, because the top and bottom edges of a character sequence are not always ideally flat. In

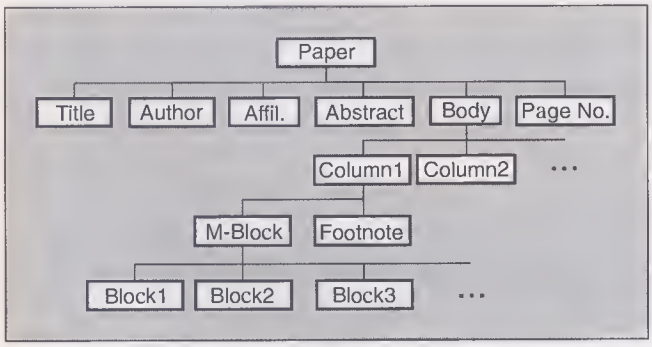


Figure 3. An example of a tree structure using the front page of an academic paper.

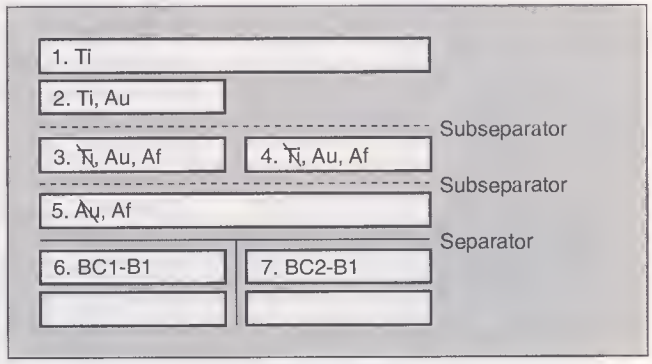


Figure 4. An example of labeling.

Table 1. Definition table for the layout model in Figure 3.

Nest	Name	Dir.	Element	Min.	Max.
0	Paper	Ver.	Dummy	1	1
1	Title	Ver.	String	1	3
1	Author	Ver.	String	1	3
1	Affiliation	Ver.	String	1	3
1	Abstract	Ver.	String	1	10
1	Body	Hor.	Dummy	1	3
2	Column	Ver.	Dummy	1	1
3	M-Block	Ver.	Dummy	1	10
4	Block	Ver.	String	1	*N*
3	Footnote	Ver.	String	1	6
1	Page No.	Ver.	Dummy	1	1

Table 2. Definition table for the labeling example in Figure 4.

Nest	Name	Dir.	Element	Min.	Max.
0	Paper	Ver.	Dummy	1	1
1	Title	Ver.	String	1	3
1	Author	Ver.	String	1	3
1	Affiliation	Ver.	String	1	3
1	Body	Hor.	Dummy	1	3
2	Column	Ver.	Dummy	1	1
3	Block	Ver.	String	1	*N*
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

this case, adjacent rectangles in the buffer are integrated into a component. In our method, constraints other than spatial adjacency (which corresponds to pixel connectivity in component labeling) can be used to determine whether two rectangles should be integrated. For example, we use the constraints that rectangles with approximately equal heights can be integrated, and that a newly integrated rectangle should not intersect other rectangles. The smearing technique sometimes connects characters to graphics or to characters in the next line (when an input image is skewed). If the connected regions are regarded as one component, it is very difficult to divide this component into its constituent parts. Use of the constraint information mentioned above can prevent incorrect integrations. The results of the integration are classified as character lines, horizontal lines, or other objects, according to the rectangle's height. For more accurate classification, it is also possible to use other features extracted from the original image.

Page structure analysis.[1] Figure 3 shows a layout model for the front page of an academic paper. Table 1 is a definition of the model. In the table, "Nest" shows the level of the tree, "Dir" indicates whether the successors of the node are arranged horizontally or vertically, "Min" shows the minimum number of successors, and "Max" shows the maximum number. "String" in the "Element" column means a terminal node (a character string), and "Dummy" means an internal node. The advantage of our method[3] is that the model can be described without rigid coordinates and procedures for structural analysis.

Horizontal and vertical separators — long and narrow white regions and black lines — are first detected on the basis of rectangles generated by block segmentation. Subseparators that represent changes of line pitch or character size are then detected. After an appropriate layout model for a page has been selected, labels for objects in the model (for example, title, author, and body) are consistently assigned to character strings.

Figure 4 shows an example of labeling, and Table 2 gives its definitions.

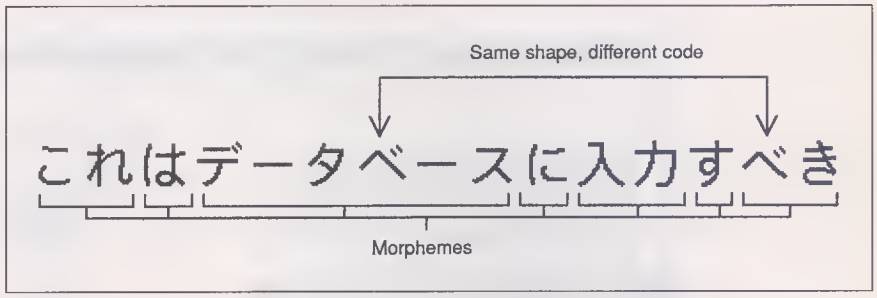


Figure 5. Japanese characters that have the same shape but different code.

Since the layout model defines the "Title" object as consisting of one to three character lines, the label Ti is assigned to character strings 1, 2, 3, and 4. Labels Au for "Author" and Af for "Affiliation" are assigned in the same way. Since separator information determines that the two-column area starts from character strings 6 and 7, neither label Au nor Af can be assigned to either of them. Therefore, the label for character string 6 is determined to be BC1-B1 (Body/Column1/Block1). When only one label is assigned to a character string, previously assigned character strings with multiple labels are investigated and contradictory labels are eliminated. Character string 5 must be assigned as Af; thus, the label Au is eliminated. Ti is similarly removed from character strings 3 and 4. If multiple labels are assigned to a character string after it has been consistently labeled, all combinations of candidate labels are investigated to calculate the confidence value from subseparator information, and the most likely path — the one that best coincides with changes of label and separator information — is determined. In this example, the most appropriate path for character strings 1, 2, (3, 4), and 5 is determined to be Ti-Ti-(Au-Au)-Af.

Text recognition. In response to requests from a workstation, the OCR processor card recognizes characters. Each character line is first segmented into characters. Each segmented character image is then normalized in size to extract feature data. The feature data consist of the distribution of local contour directions and the depths from the frame of the buffer to the nearest black pixel. The set of extracted feature data is matched with every template in a recognition dictionary, and distances are calculated to select up to five candidates with short distances.

The results from the OCR card are sent to the contextual postprocessing module to improve recognition accuracy. As Figure 5 shows, many Japanese characters have the same shape but different code. The character recognition process cannot discriminate between these characters. The postprocessing module selects a probable character sequence from a lattice of recognition candidates by using a Japanese dictionary containing about 100,000 morphemes and constraints on the transitions of morphemes. As a result, the two characters in our example are identified as parts of both a noun meaning "database" (the third morpheme in Figure 5) and an auxiliary verb meaning "should" (the last morpheme); the correct character codes are then determined. By comparing the postprocessed results with the original results obtained by the OCR card, the DRS detects uncertain characters and replaces some with lower ranked candidates.

Experiments and test results. Figure 6 shows a screen on which the front page (3,072 × 4,480 pixels) of an academic paper is being processed by the DRS in the environment described previously. Separate windows display the layout model, the page structure together with the image, and the recognized text. In the text window, the DRS warns the user about portions to be checked on the basis of results from contextual postprocessing. The user can

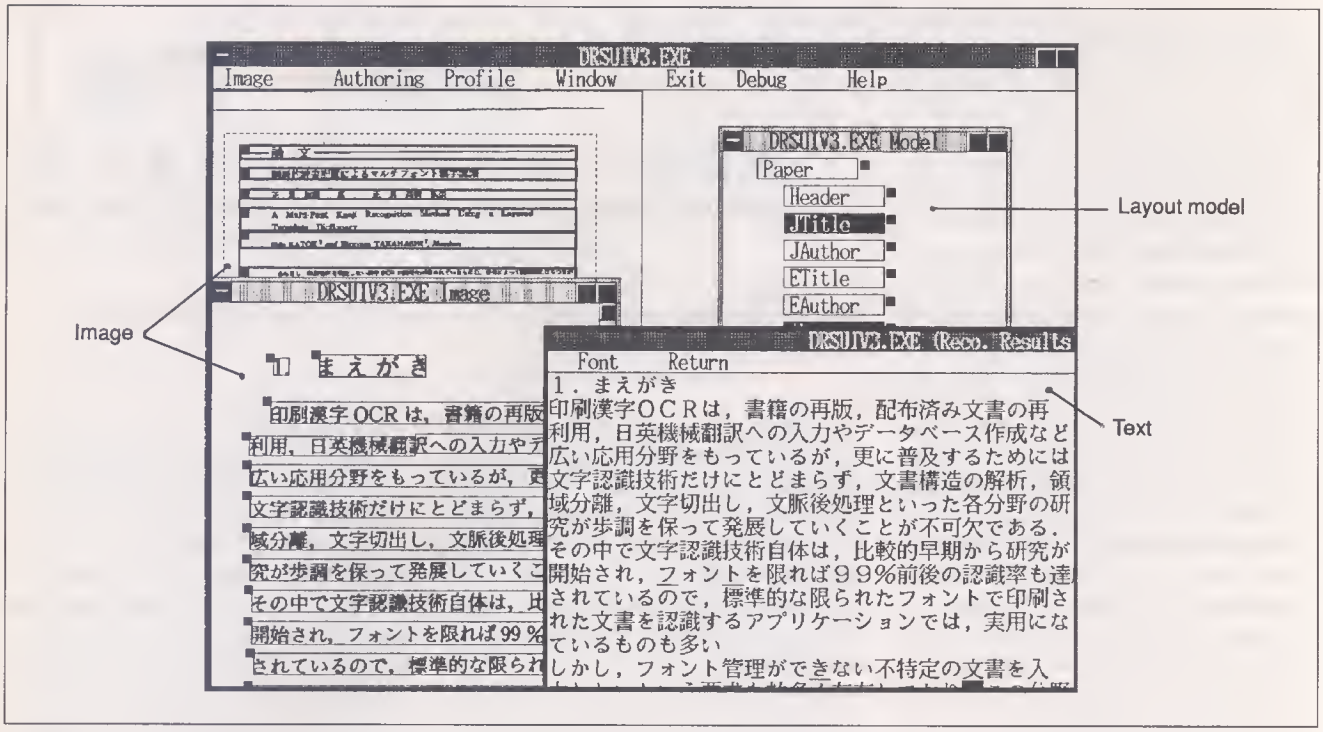


Figure 6. A screen showing the Document Recognition System at work.

correct recognition errors in this window with a mouse. The DRS takes 17-20 seconds per page to perform block segmentation (the smearing threshold was 70 pixels in length) and page structure analysis. Out of 110 images of the front pages of papers and patents, 92 pages were analyzed correctly. Character recognition and postprocessing are performed in parallel at speeds of 30 characters per second and 27 characters per second, respectively. The accuracy after postprocessing (except for superscripts and subscripts) was about 99 percent for clean pages of academic papers.

Although the current version of the DRS should be further tested and the accuracy of each function enhanced, preliminary results for several kinds of documents were satisfactory. The DRS can be incorporated easily into a standard office automation environment. Its main functions have a high processing speed and can be exploited on a workstation with OCR hardware. These functions and system services (such as a client-server mechanism and a window system) provide a flexible configuration adaptable to specific applications. In addition, software-implemented functions can be performed on the latest (fastest) hardware; dedicated hardware tends to become outdated quickly. We are continuously enhancing the DRS to make it a practical system. ■

Acknowledgments

We thank Hiroshi Maruyama and Masayuki Morohashi of the Japanese Processing Group in IBM's Tokyo Research Laboratory, and colleagues in the IBM Yamato Laboratory who developed the Japanese OCR processor card. We also thank our anonymous reviewers for providing detailed comments and constructive suggestions.

References

1. R.G. Casey and K.Y. Wong, "Document Analysis Systems and Techniques," in *Image Analysis Applications*, R. Kasturi and M.M. Trivedi, eds., Marcel Dekker, New York and Basel, Switzerland, 1990, pp. 1-35.

2. T. Amano, A. Yamashita, and H. Takahashi, "A Character-String Detection Algorithm Using Horizontal Boundaries and Its Application to a Part-Number Entry System," *Proc. SPIE*, Vol. 1,452, Image Processing Algorithms and Techniques II, Soc. of Photo-Optical Instrumentation Engineers, Bellingham, Wash., 1991, pp. 330-339.

3. A. Yamashita et al., "A Model-Based Layout-Understanding Method for the Document Recognition System (DRS)," *Proc. First Int'l Conf. Document Analysis and Recognition*, Association Francaise pour la Cybernétique Économique et Technique, Paris, 1991, pp. 131-138.

Tomio Amano is a research staff member at IBM's Tokyo Research Laboratory, where he has conducted research on OCR and block segmentation of document images. His research interests include pattern recognition, document image processing, and user interfaces.

Akio Yamashita is a research staff member at IBM's Tokyo Research Laboratory, where he has conducted research on alphanumeric OCR, parts-number entry systems, and printed Kanji OCR. His research interests are pattern recognition, image processing, and document understanding.

Nobuyasu Itoh is a research staff member at IBM's Tokyo Research Laboratory. His re-

search interests include pattern recognition, natural-language processing, and the biological mechanism of vision.

Yoshinao Kobayashi is a research staff member at IBM's Tokyo Research Laboratory. He worked as an electrical engineer before conducting research on pattern recognition — especially Kanji OCR hardware.

Shin Katoh is an advisory researcher at IBM's Tokyo Research Laboratory, where he conducts research on Kanji OCR systems and on-line recognition systems. His areas of interest are pattern recognition and image processing.

Kazuharu Toyokawa is a research staff member at IBM's Tokyo Research Laboratory, where he manages a group doing research related to image processing and pattern recognition. He has also done research on data compression.

Hiroyasu Takahashi is a research staff member with IBM's Almaden Research Center. Since joining IBM, he has performed research on operations, remote sensing, image processing, pattern recognition, document analysis, and neural networks.

Off-Line Arabic Character Recognition

Habib Goraine, Mike Usher, and Samir Al-Emami
University of Reading, Dept. of Cybernetics, Reading, Berkshire, RG6 2AY, UK

Machine character recognition has been a major object of pattern recognition research for many years, but most of the work has been on Latin, Chinese, and Indian characters. Work on Arabic character recognition started only recently.

Arabic script differs from English in several ways:

- Arabic is a cursive script that, like English handwriting, links the letters of a single word.
- It has 28 basic letters, and generally, each letter has four possible shapes depending on its position within a word.
- In addition to the 28 main letters, Arabic has special letters such as

ة , ى , لا , آ

and hamza letters such as

ء , أ , ؤ , ئ , ئ

Perhaps because of these distinctions, Arabic character recognition remained an almost untouched field until 1980, and the first approach was limited to recognizing isolated characters. Within the past five years, however, El-Sheikh and Guindi[1] developed a new method for recognizing handwritten Arabic characters that obtained a recognition rate of 93 percent. Since then, off-line recognition systems have been developed for handwritten and typewritten characters.[2] Both methods are based on the segmentation of Arabic words into strokes. Our approach also uses segmentation, but it differs in the way the strokes are classified and the number of strokes identified.

Preprocessing. Our IBM PC-based system performs three preprocessing stages sequentially: thinning, stroke segmentation, and sampling.

Thinning. We enter isolated Arabic words via a video camera and digitizer and use a thinning process to obtain characters in the form of a string of points. This gives the possibility of creating dynamic information, such as the stroke sequence, from a static image.

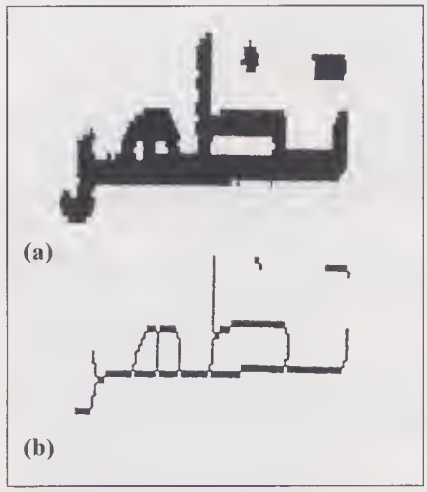


Figure 1. Binary image (a) and thinned image (b) of an Arabic word.

First, we apply Hilditch's method,[3] which consists of removing the pixels that lie on the edge of the binary image until only a one-pixel-wide line remains. This is followed by some conditions suggested by Al-Emami[2] to reduce the junction points to one junction point. Figure 1 shows a digitized image with its skeleton.

Stroke segmentation. The system segments Arabic words into principal strokes, which are strings of coordinate pairs, and secondary strokes, which are additions to the principal ones, according to the following classification:

- *connection point* — a pattern pixel that has only two neighbors;
- *feature point* — either a *line end*, a pattern pixel with only one neighbor, or a *junction*, a pattern pixel with three neighbors;
- *stroke* — a string of pixels between two successive feature points.

To follow a curve, we use a three-by-three window, as shown below, and check the eight neighboring pixels in numerical order.

n_4	n_3	n_2
n_5	p	n_1
n_6	n_7	n_8

The segmentation procedure we developed is similar to that of Al-Emami.[2] The idea of the stroke-finder algorithm is to find the start point using the flowchart given in Figure 2, then to find the end point, and finally to trace the stroke from the start to the end. Hence, the stroke is isolated and eliminated from the thinned image. The first (right-most) point detected is taken as the end point of the next stroke, and the segmentation procedure is repeated until there are no more strokes.

Smoothing and sampling. We based the sampling algorithm on a simple idea called "angular segmentation." The algorithm imposes the condition that the direction of the curve between two consecutive sampled points does not exceed a certain threshold angle. For example, Figure 3 shows the process of sampling the word

where five strokes were obtained from the segmentation process (Figure 3a).

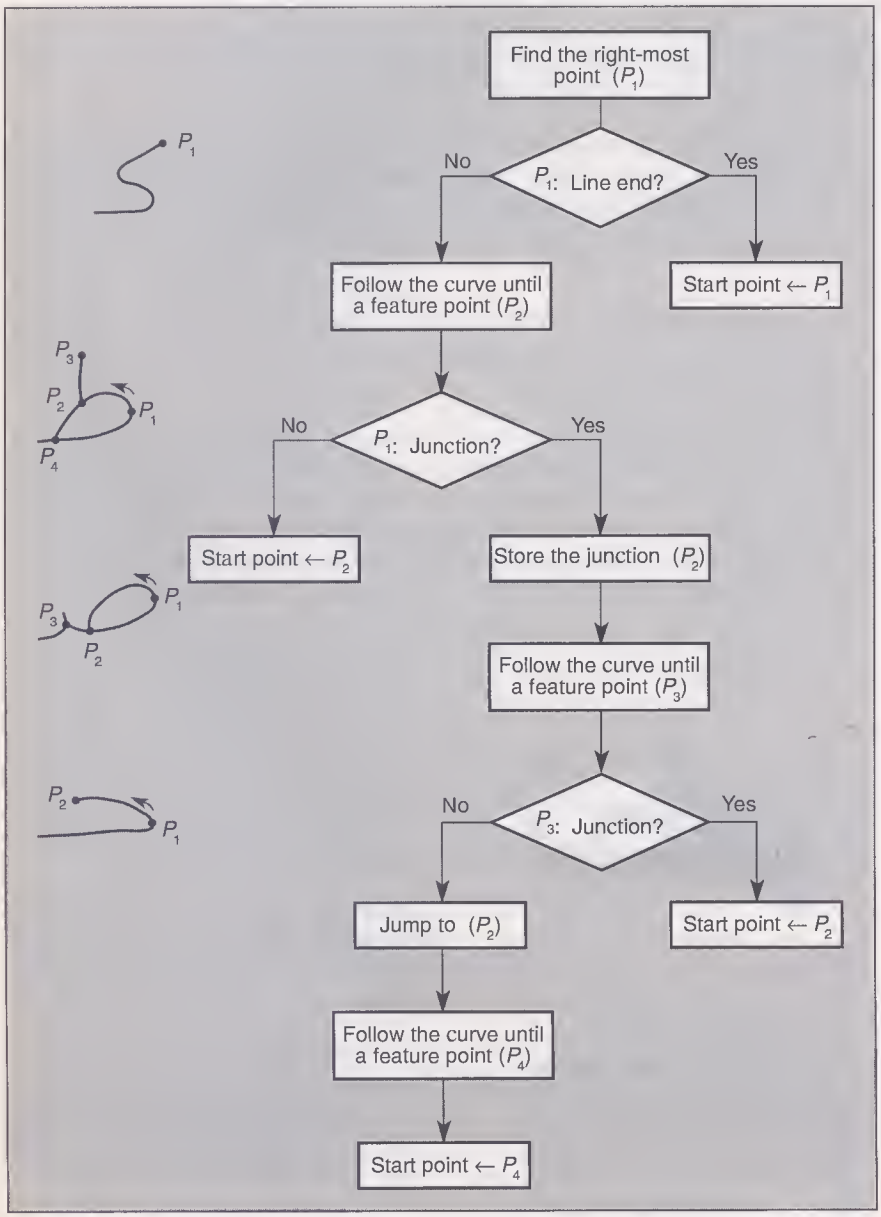


Figure 2. Flowchart for the extraction of the start point of a stroke.

Note that loops and secondary strokes have been eliminated in the sample image (Figure 3b).

Stroke representation and classification. For stroke representation, we use an eight-direction code (see Figure 4). The parameters θ_1 and θ_2 can be optimized for a particular printing style, but for our test we chose $\theta_1 = \theta_2 = 20$ degrees. The directional code sequence resulting from this quantization is then further reduced by deleting repeated codes. Each stroke is then represented by a string of characters, typically of one to nine characters.

Consider the word in Figure 3a. This word was segmented into five strokes. These strokes were coded, and after the application of some transformation rules for rounding the corners, new shapes (coded "cbah," "a," "edcbah," "a," and "dcba" from right to left, respectively) are produced. As for stroke classification, we defined 11 primitives, shown in Table 1 with their associated code sequences. Each input stroke string is matched against each reference until an exact match is found. The program has a learning capability and stores this new shape in a lookup table, which is the way the lookup table was created.

Character classification. Character classification is done at both primary and secondary levels.

Primary classification. The primary classification, which is a decision on the character or an indication of the presence of several characters or parts of a character, is based on a description of each character in the form of features. The features used can be classified as follows:

- *stroke position* (f_1) — The shape of a letter in a word is position dependent (isolated, initial, medial, terminal). A variable f_1 is assigned to the stroke position.
- *stroke shape* (f_2) — During the stroke classification, every input stroke is classified into 11 primitives (see Table 1).
- *loop in stroke* (f_3) — Most Arabic characters contain a loop, which is defined as a closed curve. A loop is identi-

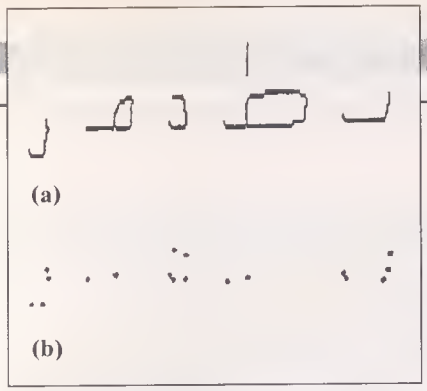


Figure 3. Sampling process: (a) strokes obtained from segmentation; (b) sampling of strokes.

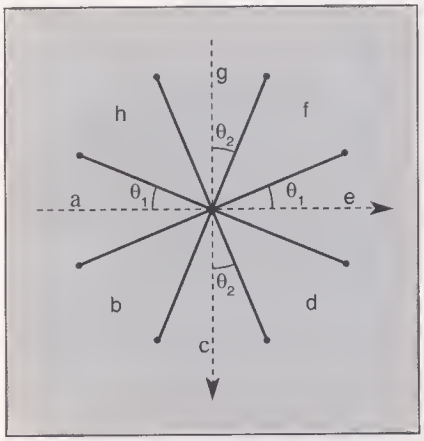


Figure 4. Eight-direction code.

fied during the segmentation procedure, and the corresponding stroke assigned a value of 1. If the stroke does not contain a loop, it will be assigned a value of 0.

• *dot number* (f_4) — During scanning of the thinned image word, the small segments (group of dots) are detected and eliminated from the image, but their midpoint coordinates and lengths are retained. Depending on its length relative to stroke width, each small segment will be considered one dot or two dots.

• *dot position* (f_5) — This is represented by the variable f_5 and is assigned a value of 1 if the group of dots is above the stroke and a value of 2 if it is below.

• *secondary stroke* (f_6) — Some Arabic characters are composed of two strokes. One is the principal stroke, and the other is a secondary stroke. If the principal stroke contains a secondary stroke, the variable f_6 is assigned a value of 1, otherwise 0.

Secondary classification. The secondary classification combines strokes into characters and solves ambiguities between pairs of characters using geometrical measurements on the character and layout context.

Among the extra features used for solving ambiguities are the following:

• *relative distance between the stroke's start point and end point* — Usually this is due to merged letters.

• *loop frame* — A loop is identified and separated from the strokes, but its frame is retained for distinguishing between different loops.

• *layout context* — This covers baseline information and location of one character with respect to its neighbors.

Postprocessing. As a final stage, for error detection and correction, system output is fed into a contextual postprocessor. This stage uses a dictionary look-up method. A search is made in the dictionary, and an input word is changed to the dictionary word if there is only one similar candidate with one single character in error; otherwise, the input word is rejected.

Experimental results. Our system consists of a hardware section for data acquisition and systems analysis and a software section to process the data to make a final decision on the character. In all the experiments, the data input device was a CCD camera and digitizer. Words were selected from the text one at a time and entered into an IBM PC computer in the form of a binary image for processing.

In the first experiment, we used four different types of printed Arabic character fonts for both training and testing. There were 200 different printed words, or about 1,200 characters in all. The recognition rate was about 92 percent.

In the second experiment, we constructed a set of cursive word images by asking three people to rewrite text printed in a font called Naskhi. They used pencil and white paper. We asked them to maintain good legibility and keep the words on a horizontal rather than slanted baseline. We used a total of 180 words comprising about 600 characters. The recognition rate was about 90 percent.

Table 1. The 11 primitives with associated code sequences.

Code Sequence or Shape	Stroke Type or Primitive	Example of Character
c		
cba		
abc		
cbahg		
edcba		
abcde		
gfedc		
abcdedcba		
edcbabcde		
ghabcbahg		
ahg		

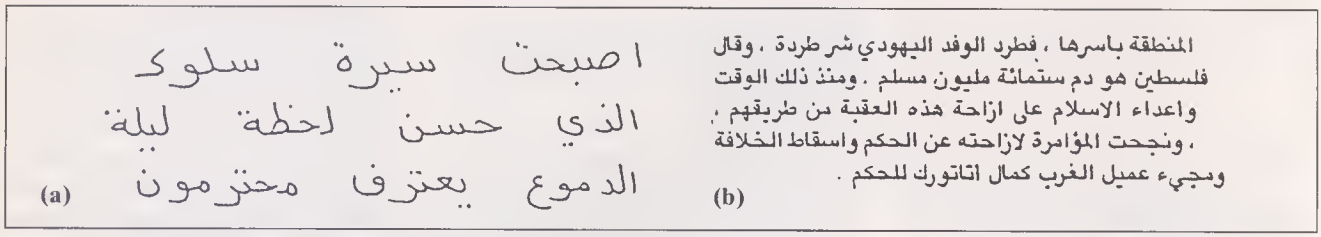


Figure 5. Samples of handwritten (a) and typewritten (b) Arabic words.

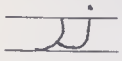
Figure 5 shows samples of both typewritten and handwritten words.

Analysis of recognition errors. As in any recognition system, some errors occurred in the experiments we conducted. We found two types of errors: substitution errors and rejection errors.

Rejection errors are usually caused by noisy images, bad printing, or handwriting that leads to bad segmentation. These errors occur when there are no reference patterns in the dictionary.

Substitution errors are the most common in our system. They usually occur because of baseline or thinning problems.

Baseline errors. In one case, a substitution error occurred even with baseline information. This particular word



consists of only one zone, namely the middle zone, so both characters appear to be above the baseline when they should be below it. Hence, they are classified as

instead of

This error could be solved at the text recognition stage where the baseline could be computed for a whole line instead of one baseline for each isolated word.

Thinning of a plain loop. The input image word

is thinned as

The loop of the first character was deleted completely. Hence, this character was classified as

instead of

This error has been corrected by using the dictionary as a spelling checker.

Conclusion. Our experiments demonstrated that the system gives high recognition for four different printed fonts and for cursive script by three writers who did not receive any handwriting training before the test. ■

References

1. T.S. El-Sheikh and R.M. Guindi, "Computer Recognition of Arabic Cursive Script," *Pattern Recognition*, Vol. 21, No. 4, 1988, pp. 293-302.

2. S. Al-Emami, *Machine Recognition of Handwritten and Typewritten Arabic Characters*, PhD thesis, Univ. of Reading, Dept. of Cybernetics, Sept. 1988.

3. C.J. Hilditch, "Linear Skeleton from Square Cupboards," *Machine Intelligence*, 4, 1969, pp. 403-420.

Habib Goraine is working on Arabic character recognition in the Department of Cybernetics, University of Reading, where he received his PhD in 1990 for research on the same topic. He obtained his BSc in Algeria.

Mike Usher joined the University of Reading's Department of Cybernetics in 1966. His interests include measurement and sensors in addition to various AI topics. He received his PhD in physics from Imperial College, London.

Samir Al-Emami returned to Iraq after completing his PhD in Arabic character recognition at the University of Reading in 1989. Before coming to the United Kingdom in 1986, he was a lecturer at the Arabian Gulf University.

Moving?

Name (Please Print)

New Address

City State/Country Zip

PLEASE NOTIFY US 4 WEEKS IN ADVANCE

MAIL TO:
IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854

ATTACH LABEL HERE

- This notice of address change will apply to all IEEE publications to which you subscribe.
- List new address above.
- If you have a question about your subscription, place label here and clip this form to your letter.

Understanding Diagrams in Technical Documents

Robert P. Futrelle, Ioannis A. Kakadiaris, Jeff Alexander, Catherine M. Carriero, and Nikos Nikolakis
Biological Knowledge Laboratory, College of Computer Science, Northeastern University, Boston, MA 02115

Joseph M. Futrelle
Hampshire College, Amherst, MA 01002

Converting documents to knowledge bases requires that the computer function as an intelligent document "reader" or "viewer." This artificial intelligence task involves computer vision and natural-language understanding. The Biological Knowledge Laboratory at Northeastern University is developing such a system. The lab's goal is to develop a knowledge base of biological research papers that supports the *Scientist's Assistant*, an intelligent system that will provide a scientist with interactive access to the research results, methods, and reasoning in a collection of scientific papers.

Document Understanding System. The system (see Figure 1) comprises a series of modules, starting with document scanning and ending with the Scientist's Assistant. The Scientist's Assistant is based on the paradigm of conceptual retrieval, which allows the user to find specific passages and data even if the user doesn't know the exact form in which the material is stored. With the Scientist's Assistant, a scientist ultimately should be able to point to a feature in a diagram from an older paper and ask, "Do people now understand the origin of this?" The Scientist's Assistant will then be able to find the most recent discussion of the phenomenon. For this to occur, the diagrams in the documents will have to be analyzed and the diagram contents added to the knowledge base.

We already know how to implement some of the modules shown in Figure 1, but implementing other modules will require extensive research and experimentation.[1,2] This means that it is not yet possible to test the entire system or even get input for some of the later modules when they depend on modules still under development. To avoid this impasse, we use alternative paths through the system during development and testing (represented by the dashed lines in Figure 1).

The novel aspects of the system include the design and use of *graphics constraint grammars* for describing and analyzing diagrams, the use of spatial indexing in diagram analysis and understanding, and extensions of natural-

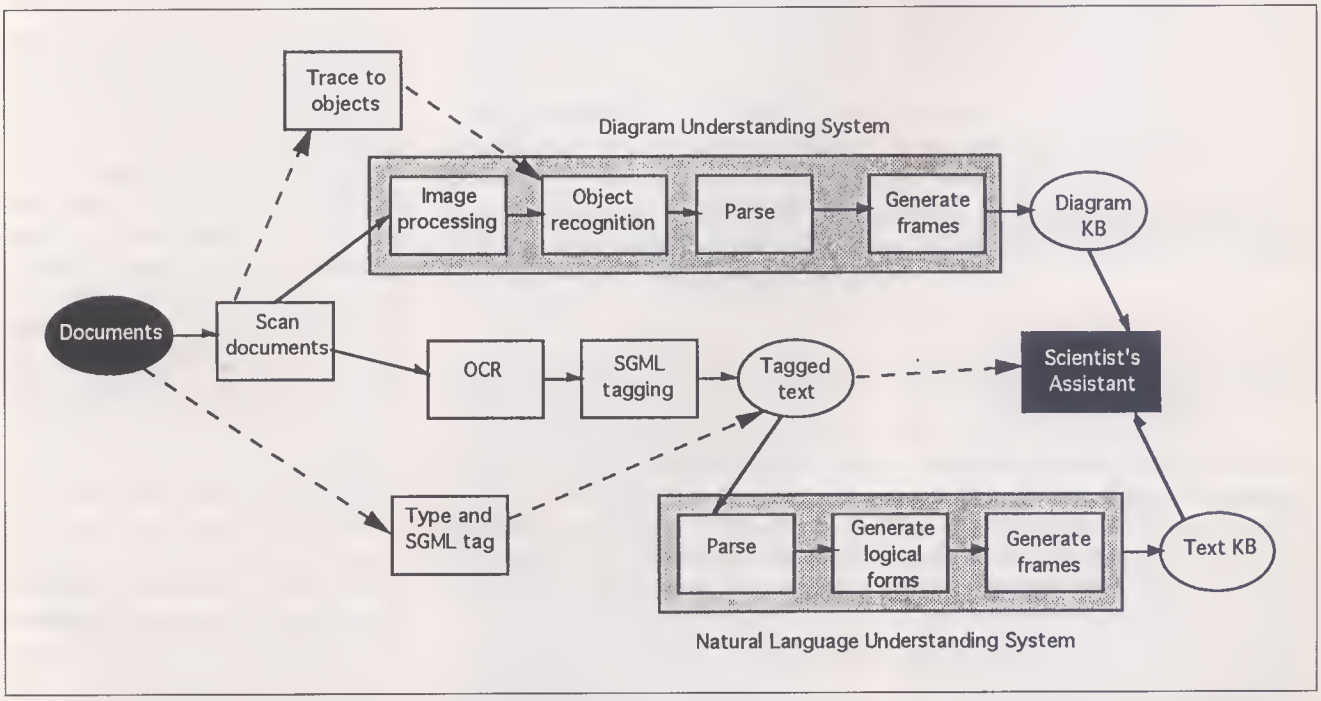


Figure 1. Overview of the Document Understanding System. Ovals denote databases and knowledge bases, rectangles represent processing systems and subsystems, and dashed arrows show the alternative strategies used during system development.

language processing techniques to complex scientific text. This article will emphasize the Diagram Understanding System. See Futrelle et al.[1] for a discussion of text processing.

The Diagram Understanding System (shown in Figure 1) entails:

Documents: Our corpus contains 1,518 papers covering essentially all of a subfield of biology (bacterial chemotaxis) from its beginning in 1965.

Object form of diagrams: Using image processing or the alternative of tracing over scanned images, the diagrams are converted from scanned, pixel-based images into a collection of graphical objects, such as lines, polygons, and positioned text.

Diagram understanding: Graphics constraint grammars are used for syntactic and semantic diagram analysis. Spatial indexing is used to rapidly discover spatial relations. The output is knowledge frames.

Tagged text: This comprises an indexed database of the entire text of each paper, encoded using the Standard Generalized Markup Language (SGML). The encoding marks every logical element such as sections, paragraphs, and sentences, as well as notations such as superscripts, subscripts, and Greek letters.

Natural-language understanding: This is a complex enterprise employing lexicons, grammars, parsers, and semantic interpreters, resulting in linked knowledge frames representing text semantics.

The Scientist's Assistant, the intelligent system that allows a scientist to navigate through the knowledge bases, is the goal of the project.

Diagram Understanding System — graphics constraint grammars. Just as we must learn the language we read and write, we must also learn the representational conventions of diagrams. Once learned, the process of reading or interpreting a diagram is relatively effortless. However, for a machine, these tasks are not so easy. A human must first describe the representational conventions formally and concisely in a way

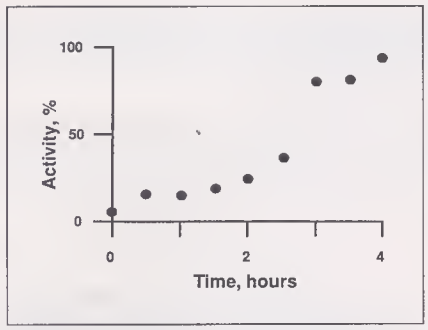


Figure 2. A data graph is the common type of diagram appearing in scientific and technical papers. The data points are the most important informational elements. The more regularly arranged elements, such as the tick marks, serve a supporting role.

that allows the computer to carry out an analysis. For diagrams, this description is similar to the grammars that are written to describe natural language.

One of the major conventions in diagrams is the separation of the informational components from the substrate on which the information is presented. For example, in Figure 2, the primary informational items are the data points. The vertical and horizontal scale lines are substrate items, serving as a "frame" in which to present the data. This division of labor can be subtle. For example, the positions of the circular data points are informational, whereas the diameters of the circles are not.

We describe the organization of diagrams with grammars. A grammar is a logical specification of a possibly infinite set of structures. It specifies a set of objects, the objects' attributes, and their relations. In the graphics constraint grammars we have developed, low-level elements are objects such as lines and polygons. High-level objects are more complex structures such as Data_points or Scale_lines. Graphics constraint grammars are similar to the approach Helm, Marriott, and Odersky[3] developed independently. The major difference in the approaches is that ours includes generalized equivalence relations and spatial indexing (both described below). Another difference between the approaches is that our grammars are incorporated in a complete system for document understanding.

Each graphics constraint grammar is a collection of rules (see Figure 3) comprising a production, a set of constraints, and a set of propagators:

- A production names the rule object as its left-hand side and the constituents of the object as its right-hand side.
- Constraints consist of spatial relations (such as Near, Horizontal, Aligned, etc.) as well as type constraints, which require that an object be of a certain type, such as a line or text.
- Propagators describe the relations between the attributes of the rule object and the attributes of the constituents. For example, the Center attribute of a set of lines might be computed as the center-of-mass of the set of lines.

The constituents of a rule may each be complex entities defined by still other rules. This allows us to build hierarchical descriptions of complex diagrams.

Example diagram and grammar. The example diagram of Figure 2 is a typical data graph. The highest level object of its hierarchical description has the constituents Data_set, Vertical_scale, and Horizontal_scale. The Horizontal _scale in turn has constituents Horizontal _scale_line and Axis_label —"Time (in hours)" in the example.

Figure 3 presents a fragment of the graphics constraint grammar for a data graph. Rule 1 defines the rule object, Horizontal_scale_line, with constituents Horizontal_axis_line and Ticks. The Ticks object is of type Labeled_x_ticks, which rule 2 defines as a pair of sets whose elements are Tick and Label, respectively. The propagator for rule 1 sets the value of the Head attribute of Horizontal_scale_line to the line object, Horizontal_axis_line. The Head attribute is the single item that best represents the rule object. The propagator for rule 2 sets the value of Head to a bounding box, the smallest rectangle surrounding all the objects in the sets Tick_marks and Label_set.

Generalized equivalence relations as constraints. One of the distinguishing

characteristics of the substrate of the data graph in Figure 2 is its simple and regular organization. For example, the *x*-axis tick marks are horizontally aligned and equally spaced. This organization is reflected in the two corresponding constraints in rule 2.

Sets of items that group together like the tick marks in the example can be described by equivalence relations. A simple equivalence relation is Equal_length. When applied to a collection of lines, Equal_length divides the lines into a collection of nonoverlapping equivalence classes, each containing lines of the same length. An equivalence relation is reflexive, symmetric, and transitive. We have extended the notion of the equivalence relation to that of the generalized equivalence relation. A generalized equivalence relation generalizes an ordinary equivalence relation in two ways:

(1) It can be approximate in nature so that it can produce classes that overlap.
(2) It has grouping relations (such as Equal-spaced) that are not normally thought of as equivalence relations.

An example of a strict equivalence relation is Coincident, referring to the positions of two objects. A generalization of Coincident is Near, a relation of great importance in diagram analysis. If two objects are near one another, they often have a logical relation, as do tick marks and their labels. Near is a generalized equivalence relation; it is not a true equivalence relation because it violates transitivity (for example, if *A* is Near *B* and *B* is Near *C*, it is not necessarily true that *A* is Near *C*). Another useful generalized equivalence relation is Strictly_near used in rule 1. Strictly_near requires that all parts of one object be near some part of another; it is not a symmetric relation.

Rule 1:

Production:
Horizontal_scale_line ⇒ Horizontal_axis_line, Ticks

Type constraints:
(Line Horizontal_axis_line)
(Labeled_x_ticks Ticks)

Geometrical constraints:
(Strictly_near Horizontal_axis_line Ticks L1)

Propagators:
Head ⇐ Horizontal_axis_line

Rule 2:

Production:
Labeled_x_ticks ⇒ Tick_marks, Label_set

Type constraints:
(Set_and_members Tick_marks Tick)
(Set_and_members Label_set Label)
(Line Tick)
(Text Label)

Geometrical constraints:
(Vertical Tick)
(Horizontally_aligned Tick_marks)
(Equal_spaced Tick_marks)
(Vertically_aligned :some Tick :every Label)
(Near :some Tick :every Label L2)

Propagators:
Head ⇐ (Bounding_box Tick_marks Label_set)

Figure 3. Two graphics constraint grammar rules that describe the horizontal scale line in the data graph shown in Figure 2. Rule 1 refers to an object of type *Labeled_x_ticks*, which is in turn defined by rule 2.

Efficient parsing of graphics constraint grammars. Solving a graphics constraint grammar problem can be an expensive computation. In finding a solution to a given rule, a number of possible assignments of objects to variables might have to be tried. This is the classic constraint satisfaction problem.[4] The usual combinatorial explosion met in these problems is mitigated by adopting the hierarchical view, which factors the problem into a set of small, independent problems that can be solved sequentially. Furthermore, the constraints that deal with the largest number of objects are typically generalized equivalence relations. These are designed to generate only solutions that include the maximal number of objects satisfying the constraint. In this way, large numbers of elements, such as the data points or tick marks in data graphs, are turned into single entities before they have to be dealt with in higher level rules.

Another potentially expensive set of computations involves geometrical relations such as Near or Aligned. For example, given an object *A*, we might need to find all objects *B* within a distance *L* from *A*, that is, satisfying (Near *A B L*). The normal method of doing this is to inspect every object in the diagram, compute its distance from *A*, and compare that to *L*.

Given the large amount of random-

access memory in modern machines, it is more efficient to do such computations by precomputing large data structures that make such computations run quickly — that is, by trading space for time.

In the Diagram Understanding System, this is done by building a pyramidal data structure.[2] Each level of the pyramid is a square array of cells representing the diagram at a different level of resolution. During the precomputation, each graphic object is examined, and a reference to the object is placed in any cell touched by or containing the object. The pyramidal data structure then operates as a spatial index.

Given a point in space, the objects at that point can be found immediately. Conversely, given any object, the cells it occupies are immediately available in a list stored in the object. The approach is general, because the same cell-based representation is used whether the objects are lines, polygons, curves, or text. Therefore, only one version of each geometrical constraint algorithm needs to be written — one that deals with cells.

Spatial indexing can then be used to efficiently compute a constraint such as (Near *A B L*). A level of the pyramid is picked on the basis of the parameter *L*. Only the cells adjacent to the cells including *A* are examined, and all the objects found in those cells are returned. The resolution of the pyramid stops well short of pixel-level resolution, so the pyramidal data structure is not particularly large — typically no larger than 128×128.

Some objects cover a lot of area, so it is inefficient to generate the number of cell references required for them. For example, as the parsing proceeds, bounding boxes to high-level objects, such as the one propagated in rule 2, might be quite large. They are stored in a different data structure, optimized for the efficient computation of constraints.

There are many complex issues in diagram parsing that we will not attempt to discuss here. For example, a diagram might have many interpretations, so information in a figure caption could be used to narrow the interpretation. Also, once the data is extracted from a data graph, further analysis is necessary to find data maxima, regions of high slope, etc., for building the knowledge representation that is to be queried.

Results. Thus far, the text of 137 articles has been encoded using SGML, and 270 diagrams have been converted to object form. The Diagram Understanding System is working; about a dozen diagrams have been analyzed with early versions of the system. For data graphs, the analysis has been able to reconstruct the datapoint values themselves. The current prototype of the Scientist's Assistant incorporates object-based diagrams — not bitmaps — and contains automatically generated hypertext links between text references and figures, tables, bibliographic items, and footnotes. Biologists have used the prototype system and given us valuable feedback. This feedback is helping to guide the ongoing task of incorporating more knowledge-based features in the assistant.

Conclusions. The technology we are developing has countless applications. The biomedical literature alone has grown by 7 million items since 1966 (as indexed in the Medline on-line information retrieval service) and is increasing at a rate of 300,000 items per year. Essentially every one of the 7 million items is available solely in hard copy, so all the techniques described here are necessary if any of this knowledge is to be converted into electronic form.

In the future, when scientific "papers" are originated, stored, and accessed purely electronically, it will still be necessary to analyze the text and diagrams in these electronic documents in order to build useful knowledge bases. The research described here is helping to prepare us for the age of fully electronic documents. ■

Acknowledgments

We thank Kent Wittenberg for insight provided during relevant discussions. This research was supported in part by National Science Foundation Grant No. DIR-8814522.

References

1. R.P. Futrelle et al., "Preprocessing and Lexicon Design for Parsing Technical Text," *Proc. Second Int'l Workshop Parsing Technologies*, Assoc. Computational Linguistics, Morristown, N.J., 1991, pp. 31-40.

2. R.P. Futrelle, "Strategies for Diagram Understanding: Generalized Equivalence, Object/Spatial Data Pyramids, and Animate Vision," *Proc. 10th ICPR, Int'l Conf. Pattern Recognition*, Vol. 1, IEEE CS Press, Los Alamitos, Calif., Order No. 2062, 1990, pp. 403-408.

3. R. Helm, K. Marriott, and M. Odersky, "Building Visual Language Parsers," *CHI 91, Conf. Human Factors in Computing*, ACM, New York, 1991, pp. 105-112.

4. J.A. Mulder, A.K. Mackworth, and W.S. Havens, "Knowledge Structuring and Constraint Satisfaction: The Mapsee Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, Nov. 1988, pp. 866-879.

Robert P. Futrelle is associate professor of computer science and head of the Biological Knowledge Laboratory at Northeastern University. From 1975 to 1985, he served on the biology faculty of the University of Illinois at Urbana-Champaign.

Ioannis A. Kakadiaris, who received his MS in computer science from Northeastern University, is a computer science PhD student at the University of Pennsylvania. His research interests include diagram understanding, foveal sensing, and parallel processing.

Jeff Alexander is a graduate student in computer science at Northeastern University, where he received his MS. He also works at the Charles Stark Draper Laboratory in Cambridge, Massachusetts.

Catherine M. Carriero is a graduate student in computer science at Northeastern University. Her research interests include diagram understanding and parallel algorithms.

Nikos Nikolakis is a PhD student in computer science at Northeastern University. His interests include image processing and diagram understanding.

Joseph M. Futrelle has worked for the University of Illinois and Symbolics Inc. He is an undergraduate at Hampshire College, Amherst, Massachusetts, concentrating in computer music.

IEEE COMPUTER SOCIETY

1992 Membership/Subscription Application

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

BENEFITS

Receive *Computer* automatically with membership. Written and refereed by experts, it features articles on the latest developments in computer technology, applications, and research, as well as survey and tutorial articles on topics covering the entire computer field.

Computer's popular departments include New Products, Product Reviews, Book Reviews, Standards, Open Channel (a reader forum), and Computer Society News.

Other membership benefits include the following:

- Discounts of up to 50% on **Computer Society Press books and videos.** Over 700 titles covering a broad spectrum of computer science topics including networking, communications, advanced systems, image processing, security, artificial intelligence, and visualization
- Discounts on registration to over 100 conferences, workshops, and tutorials each year
- Low subscription rates on special-interest periodicals
- Opportunity to participate in any of our 32 technical committees—networks of professionals with common interests in computer hardware, software, and applications
- Opportunity to participate in the development of more than 200 standards projects sponsored by the society
- Low member rates on COMPMAIL, the electronic mail service especially designed for computer professionals

Schedule of Fees

To join: see item 1, 2, or 3.
To subscribe: see item 4.

Membership dues and periodical subscriptions are annualized to, and expire on, December 31. Pay full- or half-year rate depending on date of receipt by the Computer Society as indicated below.

PRICES EXPIRE 12/31/92

	Half Year Mar 1-Aug 31	Full Year Sept 1-Feb 28
1 I don't belong to the IEEE and I want to join just the Computer Society	☐ $27	☐ $ 54
2 I want to join both the Computer Society and the IEEE*		
I reside in Region 1-6 (United States)	☐ $58.50	☐ $117
I reside in Region 7 (Canada)	☐ $53.50	☐ $107
I reside in Region 8 (Europe, Africa, or the Middle East)	☐ $53.00	☐ $106
I reside in Region 9 (Latin America)	☐ $49.50	☐ $ 99
I reside in Region 10 (Asia and Pacific)	☐ $48.50	☐ $ 97
3 I already belong to the IEEE and I want to join the Computer Society	☐ $11.00	☐ $22
IEEE Member Number _____		
4 SPECIAL-INTEREST PERIODICALS for new or current members		
	issues per year	
IEEE Computer Graphics and Applications	6 ☐ $12.50	☐ $ 25
IEEE Design and Test	4 ☐ $11.00	☐ $ 22
IEEE Expert	6 ☐ $10.00	☐ $ 20
IEEE Micro	6 ☐ $11.50	☐ $ 23
IEEE Software	6 ☐ $12.50	☐ $ 25
Transactions on:		
Computers	12 ☐ $12.00	☐ $ 24
Knowledge and Data Engineering	6 ☐ $ 9.50	☐ $ 19
Parallel and Distributed Systems	6 ☐ $ 9.50	☐ $ 19
Pattern Analysis and Machine Intelligence	12 ☐ $12.00	☐ $ 24
Software Engineering	12 ☐ $11.00	☐ $ 22
IEEE Annals of the History of Computing	4 ☐ $ 8.00	☐ $ 16

*ACM members who join both IEEE and the Computer Society may deduct $5 off the full-year rate; $2.50 off the half-year rate.

☐ Visa ☐ MasterCard ☐ American Express

Charge Card Number (Minimum Charge $10)

Mo. Yr.

Exp. Date

Checks drawn in local currency on a bank in the country of origin accepted from members in Australia, Austria, Belgium, Canada, Denmark, France, Germany, Ireland, Italy, Luxembourg, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland, UK, USA, and Yugoslavia.

Periodicals total $_____

CA, DC residents add applicable sales tax $_____

Membership fees $_____

Total $_____

CANADIAN residents add 7% GST; BELGIAN residents add 6% VAT $_____

Total enclosed $_____

I hereby make application for Computer Society membership and, if elected, will be governed by IEEE's and the society's constitutions, bylaws, and statements of policies and procedures.

MAILING ADDRESS

Full signature _____ Date _____

☐ Male ☐ Female

First name _____ Middle initial(s) _____ Last name _____ Date of birth _____

Street address _____ City _____ State/Country _____ Zip _____

EDUCATION (highest level completed) Course _____ Degrees received _____ Date _____

Name of educational institution _____

OCCUPATION Title or Position _____

Firm name _____

Firm address _____

City _____ State/Country _____ Zip _____

ENDORSER (an IEEE member, Senior Member, or Fellow)

Endorser's signature _____

Name (print in full) _____ IEEE Member No. _____

Street address _____

City _____ State/Country _____ Zip _____

Mail or fax to appropriate Computer Society office:

EUROPE: 13, Avenue de l'Aquilon, B-1200, Brussels, BELGIUM. Fax: 32-2-770-8505
PACIFIC RIM: Ooshima Building, 2-19-1 Minami-Aoyama, Minato-ku, Tokyo 107, JAPAN. Fax: 81-3-3408-3553
ALL OTHERS: 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720–1264 USA. Fax: 714-821-4010

PC0792

ADVERTISER INDEX

Alsys, Inc. 9

CACI Products Company 1

Compsac '92 . . . . . Cover IV

Dependable Computing for Critical Applications Conference 105

Duke University 90

Frontiers of Massively Parallel Computation Symposium 4

GTE Laboratories Inc. 54

IBM Corporation 33

ICPR '92 22

IEEE Computer Society Membership 79

Institute for Defense Analyses 95

Local Computer Networks Conference . . . . . Cover II

OOPSLA '92 43-45

Systems Management Workshop 23

Victoria University 22

The VLDB Journal 55

VLSI Test Symposium . . . . . Cover III

Classified Advertising 106-109

FOR DISPLAY ADVERTISING INFORMATION, CONTACT:

District Manager: D. Rodney Brooks; tel: (415) 905-0260; fax: (415) 896-1512.

Recruitment & Classified Advertising: D. Rodney Brooks; tel: (415) 905-0260; fax: (415) 896-1512.

Director of Sales: Randall L. Stickrod, South Park Media, 544 Second St., Suite 200, San Francisco, California 94107; tel: (415) 905-0260; fax: (415) 896-1512.

For production information, conference, and classified advertising, contact Heidi Rex or Marian Tibayan.

COMPUTER, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, California 90720-1264; tel: (714) 821-8380; fax (714) 821-4010.

PRODUCT INDEX

	RS#	PG#
A/Soft Development	40	82
Abbott Systems	39	82
ACC Microelectronics	121	91
Accton Technology	141	92
Adobe Systems	37	82
Alsys, Inc.	1	9
Alta Technology	142, 143	92
AMI	122	91
Ampro Computers	144	92
Anadigics	123	91
Apple Computer	38	82
ATI Technologies	124	91
Beckman Instruments	42	82
CACI Products Company	—	1
Comm. Automation & Control	145	92
Comtrade	22	88
Cyrix	125	91
Elographics	45	83
Heurikon	146	92
HNC Inc.	35	81
Iomega	147	92
Jandel Scientific	46	83
Keydata International	43	83
Linear Technology	126	91
Meridian Software Systems	24	90
Mitsubishi	127	91
Momentum Data Systems	44	83
Motorola	128	91
Myriad Enterprises	41	82
National Semiconductor	129	91
PEP Modular	148	92
Philips Semiconductors	129	91
Radstone Technology	149	92
Sealevel Systems	150	92
Softac Corp.	36	81
StructSoft	21	87
SunSelect	23	89
Texas Instruments	151	92
Trident	131	91
VLSI Technology	132	91
Voice Connexion	152	92

COMPUTER

Reader Service Card

July 1992 issue (Card void after January 1993)
Print address below.

Name _____

Title _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____ Phone _____

Reader interest
(Circle what you liked; add comments on the back)

- **A** Guest Editors' Introduction
- **B** Image Analysis System
- **C** Interpretation System
- **D** Postal Address Block Location
- **E** Celesstin
- **F** Project Overviews
- **G** New Products
- **H** Product Reviews
- **I** IC Announcements
- **J** Microsystem Announcements
- **K** Computer Society News
- **L** Update
- **M** Standards
- **N** Call for Papers
- **O** Calendar
- **P** Book Reviews
- **Q** Open Channel

Product information
(Circle numbers for products and advertisements on which you want more information)

1	21	41	61	81	101	121	141	161
2	22	42	62	82	102	122	142	162
3	23	43	63	83	103	123	143	163
4	24	44	64	84	104	124	144	164
5	25	45	65	85	105	125	145	165
6	26	46	66	86	106	126	146	166
7	27	47	67	87	107	127	147	167
8	28	48	68	88	108	128	148	168
9	29	49	69	89	109	129	149	169
10	30	50	70	90	110	130	150	170
11	31	51	71	91	111	131	151	171
12	32	52	72	92	112	132	152	172
13	33	53	73	93	113	133	153	173
14	34	54	74	94	114	134	154	174
15	35	55	75	95	115	135	155	175
16	36	56	76	96	116	136	156	176
17	37	57	77	97	117	137	157	177
18	38	58	78	98	118	138	158	178
19	39	59	79	99	119	139	159	179
20	40	60	80	100	120	140	160	180

COMPUTER

Reader Service Card

July 1992 issue (Card void after January 1993)
Print address below.

Name _____

Title _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____ Phone _____

Reader interest
(Circle what you liked; add comments on the back)

- **A** Guest Editors' Introduction
- **B** Image Analysis System
- **C** Interpretation System
- **D** Postal Address Block Location
- **E** Celesstin
- **F** Project Overviews
- **G** New Products
- **H** Product Reviews
- **I** IC Announcements
- **J** Microsystem Announcements
- **K** Computer Society News
- **L** Update
- **M** Standards
- **N** Call for Papers
- **O** Calendar
- **P** Book Reviews
- **Q** Open Channel

Product information
(Circle numbers for products and advertisements on which you want more information)

1	21	41	61	81	101	121	141	161
2	22	42	62	82	102	122	142	162
3	23	43	63	83	103	123	143	163
4	24	44	64	84	104	124	144	164
5	25	45	65	85	105	125	145	165
6	26	46	66	86	106	126	146	166
7	27	47	67	87	107	127	147	167
8	28	48	68	88	108	128	148	168
9	29	49	69	89	109	129	149	169
10	30	50	70	90	110	130	150	170
11	31	51	71	91	111	131	151	171
12	32	52	72	92	112	132	152	172
13	33	53	73	93	113	133	153	173
14	34	54	74	94	114	134	154	174
15	35	55	75	95	115	135	155	175
16	36	56	76	96	116	136	156	176
17	37	57	77	97	117	137	157	177
18	38	58	78	98	118	138	158	178
19	39	59	79	99	119	139	159	179
20	40	60	80	100	120	140	160	180

COMPUTER

Reader Service Card

July 1992 issue (Card void after January 1993)
Print address below.

Name _____

Title _____

Company _____

Address _____

City _____

State/ZIP _____

Country _____ Phone _____

Reader interest
(Circle what you liked; add comments on the back)

- **A** Guest Editors' Introduction
- **B** Image Analysis System
- **C** Interpretation System
- **D** Postal Address Block Location
- **E** Celesstin
- **F** Project Overviews
- **G** New Products
- **H** Product Reviews
- **I** IC Announcements
- **J** Microsystem Announcements
- **K** Computer Society News
- **L** Update
- **M** Standards
- **N** Call for Papers
- **O** Calendar
- **P** Book Reviews
- **Q** Open Channel

Product information
(Circle numbers for products and advertisements on which you want more information)

1	21	41	61	81	101	121	141	161
2	22	42	62	82	102	122	142	162
3	23	43	63	83	103	123	143	163
4	24	44	64	84	104	124	144	164
5	25	45	65	85	105	125	145	165
6	26	46	66	86	106	126	146	166
7	27	47	67	87	107	127	147	167
8	28	48	68	88	108	128	148	168
9	29	49	69	89	109	129	149	169
10	30	50	70	90	110	130	150	170
11	31	51	71	91	111	131	151	171
12	32	52	72	92	112	132	152	172
13	33	53	73	93	113	133	153	173
14	34	54	74	94	114	134	154	174
15	35	55	75	95	115	135	155	175
16	36	56	76	96	116	136	156	176
17	37	57	77	97	117	137	157	177
18	38	58	78	98	118	138	158	178
19	39	59	79	99	119	139	159	179
20	40	60	80	100	120	140	160	180

Editorial comments

I liked: _____

_____

_____

_____

I disliked: _____

_____

_____

I would like to see: _____

_____

_____

_____

For reader-service inquiries, see other side.

PLACE
POSTAGE
HERE

PO Box is for reader-service cards only.

COMPUTER
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA

Editorial comments

I liked: _____

_____

_____

_____

I disliked: _____

_____

_____

I would like to see: _____

_____

_____

_____

For reader-service inquiries, see other side.

PLACE
POSTAGE
HERE

PO Box is for reader-service cards only.

COMPUTER
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA

Editorial comments

I liked: _____

_____

_____

_____

I disliked: _____

_____

_____

I would like to see: _____

_____

_____

_____

For reader-service inquiries, see other side.

PLACE
POSTAGE
HERE

PO Box is for reader-service cards only.

COMPUTER
Reader Service Inquiries
PO Box 16508
North Hollywood, CA 91615-6508
USA

NEW PRODUCTS

Contact or send releases to Christine Miller, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264; Internet, s.c. miller @ compmail.com.

WINDOWS II, WINDOWS II, WINDOWS II

Learns from relationships

HNC Inc. has released a database modeling and analysis system called the DataBase Mining Workstation (DMW), which learns key relationships underlying historical data and generalizes from them to produce predictive models.

The KnowledgeNet facility explains the model's decisions, ranks important factors contributing to a decision, and lets users perform "what-if" queries.

The company states that DMW is particularly adept at modeling dynamic nonlinear relationships and can cut development time while allowing for greater exploration and refinement of results. The program can be used for such applications as diagnosing circuit board failures and predicting PVC (polyvinyl chloride) plastic quality and consistency.

DMW runs under Windows 3.0 and 3.1 and consists of neural network software and HNC's 80-Mflops Balboa coprocessor. It costs $19,950.

Reader Service Number 35

Escape from your mouse

The Keyboard User Interface (KUI) from Softac Corp. provides hot keys for launching and switching applications and keyboard-based mouse simulation that lets users move the cursor by using arrow keys.

Users can work from the DOS command line through an interface that is integrated with a pop-up control panel. Internal DOS commands include Copy, Rename, Delete, and Check Directory. The KUI window displays assigned hot-key names and lists keys assigned to each project.

Cursors come in small, medium, and large sizes in black or white for higher visibility on SVGA and LCD displays. Users can also magnify WYSIWYG text with the program's Lens feature.

KUI includes animated screen savers, a file-search command for multiple drives, and window-management commands that minimize or close all windows, or display two windows side by side.

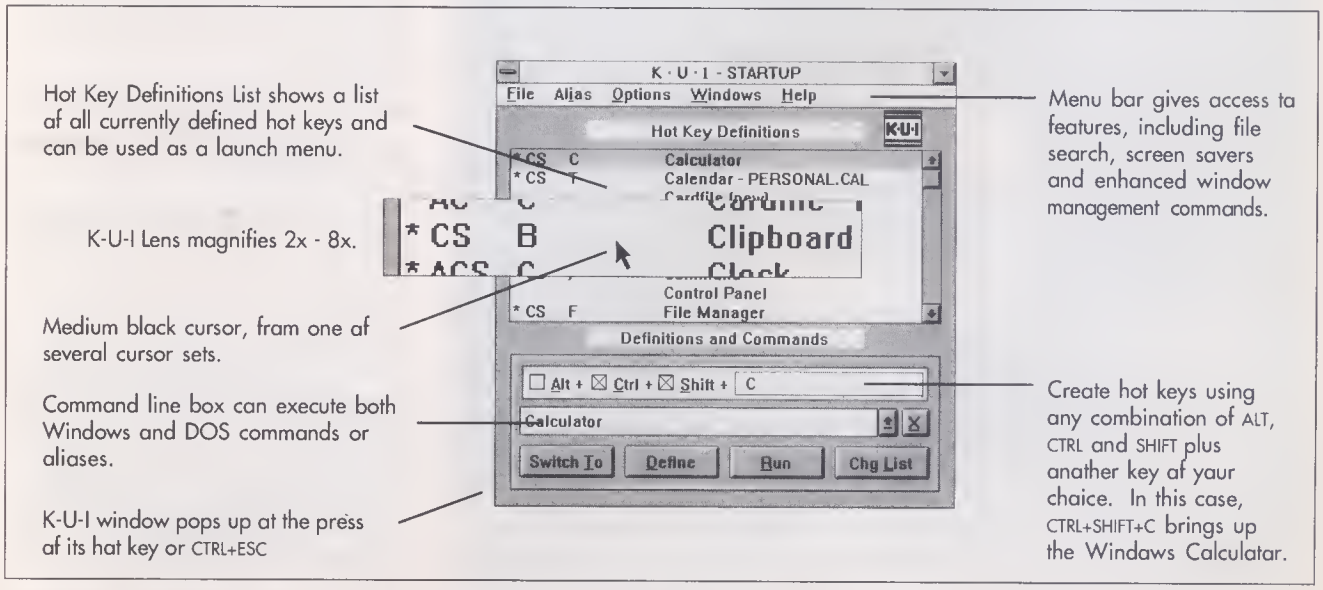
The $79.95 program requires Windows 3.0 and DOS 3.1; runs in real, standard, and 386-enhanced modes; and uses 330 Kbytes of free memory and 40 Kbytes of RAM.

Reader Service Number 36

Update your illustrator

Adobe Systems has announced that Adobe Illustrator 4.0 for MS Windows is now shipping. New features include editing in preview mode; on-screen text entry and manipulation; advanced graphing capabilities; improved color selection; on-line context-sensitve help; and extensive import and export file support.

The MS Windows version includes Adobe Type Manager, TypeAlign, and Separator programs. An optional Streamline 3.0 for Windows package converts color and black-and-white bitmapped images into PostScript line art for use in the package.



The Keyboard User Interface adds a number of features to the Windows environment.

Illustrator 4.0 requires a 386 or 486 computer, 4 Mbytes of RAM, DOS 3.3 and Windows 3.0 or 3.1, a VGA or an SVGA 16-bit or 24-bit display adapter, and a Windows-support graphic output device.

The program costs $695. Upgrades are $99, and a $199 package is available to owners of some other drawing packages. A shared file format among the Windows, Macintosh, and Next versions of the program provides cross-platform compatibility, according to the company.

Reader Service Number 37

Mixed-environment strategy

Apple Computer has introduced the OneScanner for Windows. The one-step device provides gray-scale scanning with 256 levels, adaptive calibration, and selection and scaling tools.

Ofoto application software straightens images and accommodates large image files by expanding system memory with hard disk space. The package supports TIFF, PC Paintbrush, Windows bitmap, GEM Image, Microsoft Paint, and EPS file formats. Brightness and contrast changes can be monitored on screen. Context-sensitive help is furnished.

The flatbed scanner scans 8.5 × 14-inch documents at a resolution of 72 to 300 dpi and outputs the file at 1 to 16,000 dpi.

OneScanner for Windows includes scanner hardware, a SCSI board for the PC, a cable, and the Caere OmniPage Professional trial version for $1,299.

Reader Service Number 38

Publish your calulations

Abbott Systems has announced the Calc+ calculator for use with Windows 3.1 spreadsheet and desktop publishing software. Features include a resizable display that lets you scroll through calculations and choose fonts and type size, point-and-click recalculations for spreadsheets, and conversions between picas/points, feet/inches, and centimeters.

Users can add and edit comments on a line-by-line basis.

Calc+ costs $79 and comes with free unlimited 800-line support.

Reader Service Number 39

Edit across the Windows

A/Soft Development has announced Nu/TPU Version 3.0 fully programmable text editor for MS Windows. Programmers can work in MS Windows, Motif, and OpenWindows with four built-in interfaces: Eve, EDT, WPS, and vi. Each interface uses a key-pad layout that allows common editing commands to be performed by pressing a key.

In addition to the traditional command line, a mouse, menu bar, and pop-up windows provide access to 180 commands to control multiple windows and buffers, change edit session parameters, manipulate text, control subprocesses, and define new interface functionality. Users can add editing features interactively and create new editing interfaces.

The package features 58 new editing commands such as Rectangular Cut and Paste, which can be used in both insert and overstrike mode. End users can create their own keyboard maps by editing a text file.

DOS pricing begins at $325, Unix at $499.

Reader Service Number 40

When time is of the essence

The Winsprint printer controller from Myriad Enterprises accelerates printing of Windows 3.0 and 3.1 applications on HP LaserJet II and III printers. According to the company, Winsprint halves the output time for such scalable-font applications as word processing and spreadsheets and can speed printing as much as six times for overhead transparencies. It employs PC and system memory to process, rasterize, and compose the page.

The controller consists of an interface board that fits into the optional input/output slot in the back of the printer, a 16-bit ISA interface board that installs in the PC, and cable. The printer driver may be installed as the default printer.

Winsprint requires a 386 or 486 with one open 16-bit expansion slot and at least 4 Mbytes of system memory. It costs $399.

Reader Service Number 41

Automating the lab

The EasyLIMS 2.1 laboratory information-management system from Beckman Instruments lets users log routine and nonroutine samples or requests into the system. It monitors the status of all samples being processed by the lab, prints labels and provides cost accounting, tracks instrument calibration requirements, and includes an electronic user notebook.

Results can be recorded manually or imported automatically into the package. A report generator presents an electronic page to the user, who selects text and database fields from an iconized status bar. The user then points to the selected item and paints the field with a mouse click.

The program maintains a system of user-created data tables containing information relating to product or sample types, test definitions, instruments, customers, and users. A browse function lets users search for any relevant field in the system. Using standard SQL operators, users may define the scope and content of what is presented on the browse screen.

EasyLIMS comes in single-user or networked versions and is completely integrated with Windows 3.1.

Reader Service Number 42

WINDOWS II, WINDOWS II, WINDOWS II

Machine for Windows

Keydata Int'l has announced the WindowStation computer, especially built for Windows-based applications. The computer was built from the motherboard up to eliminate 16-bit 8-MHz I/O. The Intel 486-based KEY486 uses a 32-bit local bus and a Windirect S3 SVGA accelerator board to support video performance. A 17-inch View-Sonic noninterlaced SVGA color monitor provides room for a number of windows, 1,280 × 1,024 resolution, and a 76-Hz refresh rate.

WindowStation is compatible with OS/2 and Novell. A double-clock chip configuration includes 8 Mbytes of RAM; 246-cache RAM; TEAC 1.2-Mbyte and 1.44-Mbyte drives; one parallel and two serial ports; and a 340-Mbyte 15-ms hard-disk drive.

The $3,995 workstation includes a 130-key click keyboard with a built-in calculator, MS DOS 5.0, MS Windows 3.0 (3.0 users will be upgraded to 3.1 at no charge), a high-resolution mouse, a built-in 9,600/2,400-baud fax/modem, and a choice of full- or mid-tower configuration.

Reader Service Number 43

DSP for 3.0

Momentum Data Systems has announced DSPworks 2.0 for Windows. The signal-analysis package provides signal generation, real-time data acquisition, and general signal-processing functions. It is designed to be seamlessly integrated with the company's QEDesign 1000 and Code Generators.

General features include a real-time spectral analyzer and oscilloscope, tiled and stacked graphic displays, 2D and 3D spectrograms, and extensive prototyping and simulation capabilities. Users can create sine, triangular, unit step, square wave, impulse, response, and arbitrary waveform signals. Noise can be added based on probability density functions.

DSPworks is available on MS Windows 3.0 or higher for $495.

Reader Service Number 44

3D graphs on 3.0

SigmaPlot 5.0 from Jandel Scientific lets users create mesh and scatter plots in 3D. Graphs can be interactively rotated by using a sparse matrix representation method that allows users to see the graph as it is being adjusted.

Researchers can convert *x*, *y*, *z* scatter data to the 3D mesh format by using an Interpolate Mesh command. Other options include hidden-line removal, filled or unfilled polygons, frame lines, backplanes with color and grids, and drop lines for scatter plots. Users can select the current graph only to speed up page-redrawing time while resizing, editing, and rescaling graphics. A file-merging command merges user-compiled lists of SigmaPlot graphs for resizing and scaling.

The package operates under Windows 3.0 as a full-screen application, within DESQview, and as a single-user application in Novell and other DOS-compatible networks. A graphics adapter and 3 Mbytes of free memory are required. Import files include ASCII, Lotus 1-2-3, Symphony, Quattro, Microsoft Excel, and dBase. The package costs $495.

Reader Service Number 46

Just tap on the Window

Elographics has released MonitorMouse for Windows 3.1, which installs through the control panel to let an Elographics touch screen work simultaneously with a mouse or independently. All normal Windows 3.1 functions such as resizing windows, moving objects, and using pull-down menus can be performed by touching the screen. To perform the double-click function, the user taps the screen twice. The device is compatible with serial, PC-Bus, and Micro Channel touch-screen controllers.

Reader Service Number 45

SigmaPlot exports CGM, EPS, HPGL, and DXF formats for graphics.

Send review submissions to Product Reviews Editor: J.M. Jagadeesh, College of Pharmacy, Ohio State University, Columbus, OH 43210

Powerful CASE on the Mac

— *Jack R. Hagemeister and Paul W. Oman, University of Idaho*

TurboCASE Version 4.0 is a complete front-end analysis and design tool that combines popular methodologies with the most user-friendly interface available. Under the guise of being a "small" (individual user) CASE tool, TurboCASE provides

- conventional dataflow analysis,
- real-time dataflow analysis,
- object-oriented analysis,
- structured design,
- object-oriented design,
- organizational development,
- flowcharting, and
- architectural development.

All development components are integrated with a data dictionary and can be applied to any development process or paradigm. This tool from StructSoft operates on Apple Macintosh computers with 1 Mbyte of memory. We tested it on a Classic with 4 Mbytes of memory and on a IIci with 8 Mbytes of memory and an RGB color monitor.

Working with TurboCASE

The first question most engineers and managers have about any development tool is, "How quickly can the tool be used productively?" The user interface of TurboCASE answers this concern right away by providing immediate productivity. The interface consists of the usual Mac-like menu bar across the top of the screen that outlines various options. Commands include File, Edit, Modeling Options, Check, Font Selection, Window Control and Placement, Interface, Help Index, and Others. Each option is accessed by moving the cursor to the desired item and pressing the mouse button. When the submenu of the chosen set appears, you select the command that you want and let up on the mouse switch to execute it.

The development activities are accessed through diagram windows that represent

- the various analysis or design diagrams that will be produced,
- query windows for working with text information or checking the development, and
- input windows for such textual information as names of objects or data dictionary entries.

There are also boxes for making choices when you initiate a command or need I/O assistance. Interaction in all aspects of the program is consistently implemented and well thought out. The menus and actions available in an analysis dataflow diagram are the same as those for object-oriented design and flowcharting tools. Each activity that you perform is assisted by dialogue boxes and input prompts. To create a diagram like a DFD (dataflow diagram), you start by choosing the modeling option from the menu bar, then Functional, and then New in the small box that appears. A new diagram box is created with a name that you supply. Or you can create a diagram from within another diagram by placing the cursor on the object you wish to model, holding down the mouse, and choosing Model. A new diagram is automatically created to further expand an existing object or diagram.

There are three context-sensitive pop-up menus within a drawing diagram. The first contains commands associated with the diagram as an entity. It is opened by clicking on the diagram name (see Figure 1). The second contains drawing commands within the diagram such as Create or Undelete. It is opened by pressing the mouse button when the cursor is on any blank space in the diagram (see Figure 2). The third major menu is associated with objects that have been created in a diagram. The commands in this menu pertain to the physical appearance of the object and relational aspects of the object to the development (see Figure 3).

In general, it is easiest to think of all parts of the TurboCASE system and the development project as objects. The tool can perform actions on these objects to represent a developer's work. Throughout TurboCASE, you manipulate objects in an intuitive manner by moving the mouse cursor to the object, pressing and holding the mouse button, selecting the desired operation from the menu that appears, and releasing the button. In our labs, this user interface has been introduced to several people with widely ranging familiarity with Macintosh or CASE tools, and each one could quickly use the tools and features within the program.

Supported methods

This package supports several development methodologies and provides significant assistance for both analysis and design activities. It supports Yourdon/DeMarco structured analysis, Gane/Sarson structured analysis, McMeniman/Palmer essential systems analysis, Chen entity relationship modeling, Hatley/Pirbhai real-time modeling, Ward/Mellor real-time modeling, ESML (Extended System Modeling Language) real-time modeling, Yourdon/Constantine structured design, Shaler/Mellor object-oriented analysis, and Booch-like object-oriented design.

To establish the working methodol-

We're going to miss you

Every once in a while an individual comes along who does the really special thing, does it very well, and continues to do so for a long time. One such individual is Richard Eckhouse, who is leaving his post of *Computer* Product Reviews editor, although (happily) he will still contribute to this column.

All of us at *Computer* would like to take this opportunity to thank Dick for his enormous contribution and wish him the best of luck.

ogy for a project, you select the Modeling command from the menu bar, which presents you with one active choice: Configure Functional. You then are presented with a selection box for real-time methods. The choices are None, Hatley, ESML, and Yourdon. Then you are presented with a second selection box to determine the type of DFD symbology you wish to use — Yourdon/DeMarco or Gane/Sarson. When a new project is created, the methodology follows from your choices. If you don't go through this process, TurboCASE defaults to the Yourdon/DeMarco methodology with Yourdon real-time extensions.

To start a new development project, you choose New Project from the File item in the menu bar and are prompted for a project name. TurboCASE uses this name to group all diagrams and data-dictionary information as one unit. When you continue work on an existing project by choosing the Open Project command, you are provided with an interactive box consistent with other Mac Finder dialogues that allow access to the hard drive and floppies.

Structured analysis tools. The beginning of a structured analysis is the Context Diagram. Only one process can be defined at that level. (The real-time methods do not have this restriction.) Data and control flows can be entered as needed. While you develop the diagram, TurboCASE is adding the information from the diagram to a data dictionary for the project, which reduces the burden on the analyst. The tool also uses this information for its consistency- and completeness-checking phases.

Once the Context Diagram is created, you can facilitate decomposition by creating child diagrams of the defined process. You move the cursor to the process bubble, hold down the mouse switch, and select Model. A new diagram appears with the name of the process it is modeling. Or you can double-click on a process to open the child diagram. Within this diagram, processes can be decomposed to the subprocesses necessary to define the problem.

Creating processes or data stores in any diagram is accomplished by moving the mouse cursor to a blank space and clicking the button. When an icon palette pops up, you just click on the appropriate icon. TurboCASE prompts you for a name and places the object (process or store) where you clicked. By clicking within a pro-

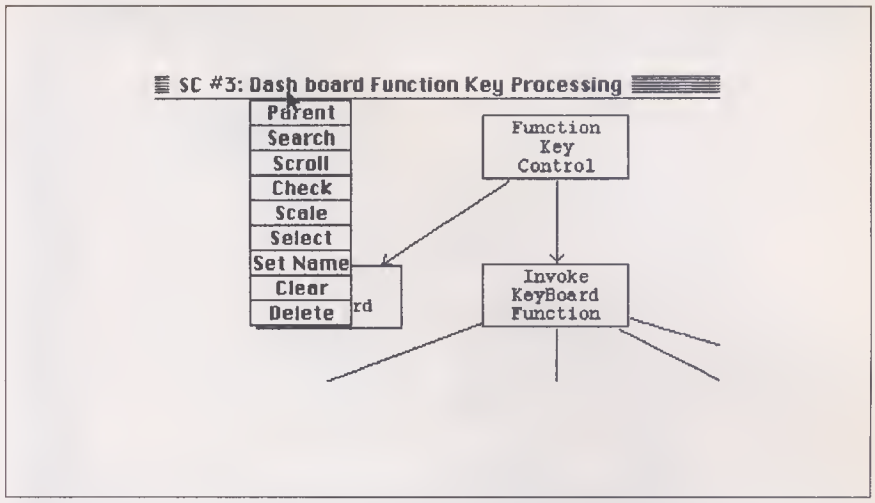


Figure 1. The diagram pop-up menu in a structure chart diagram.

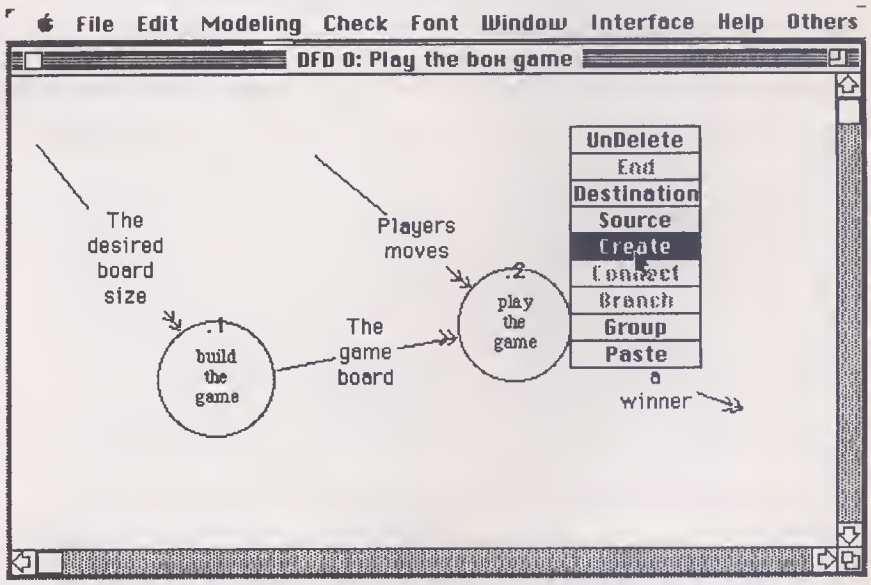


Figure 2. The pop-up menu for the drawing area in a DFD diagram.

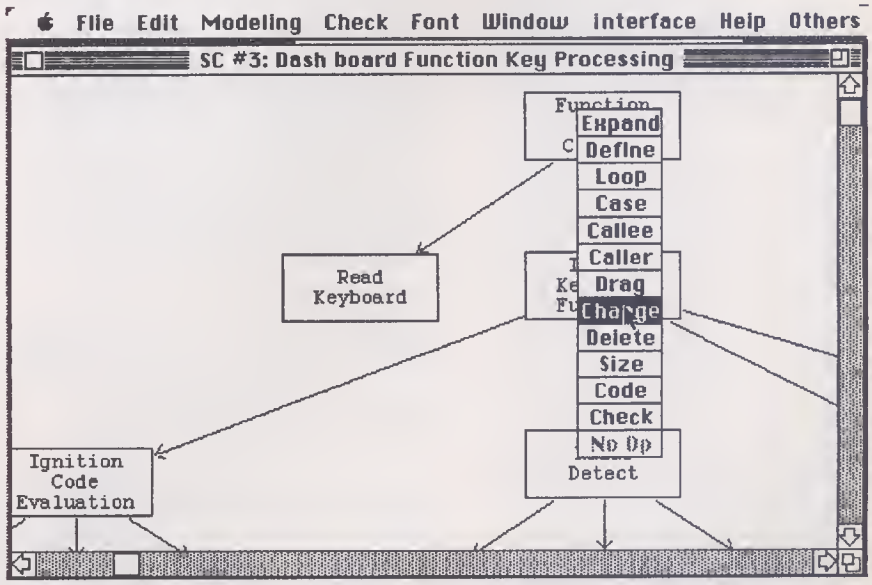


Figure 3. The pop-up menu from within a drawing object.

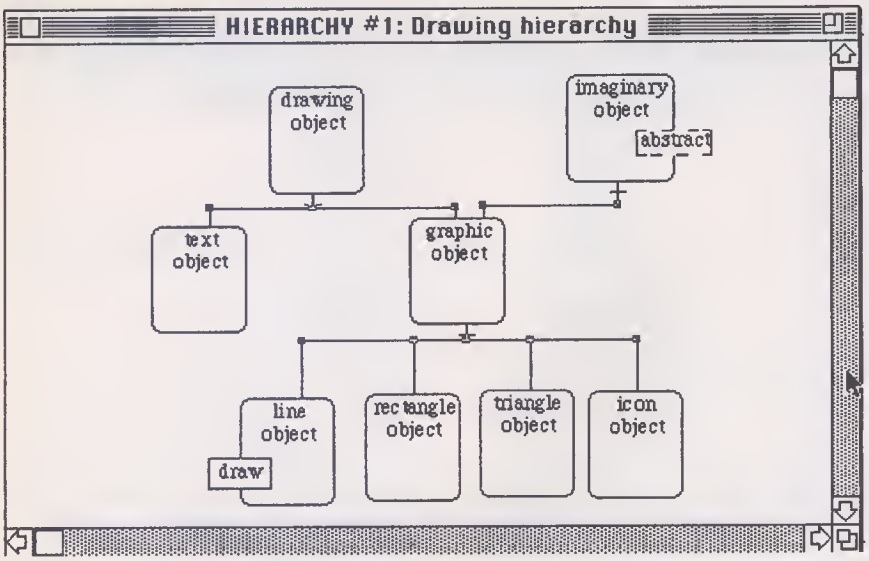


Figure 4. A class hierarchy diagram.

cess, you can access the flow connections and place dataflow or control-flow paths within the diagram. These flows are also named and added to the project database. When dataflows exist in a parent diagram, the program supplies a list so that you don't have to remember their names. Just use the mouse to double-click on the correct flow to add it to the child diagram. Four types of flows for structured analysis are provided: control prompts, event flows, intermittent dataflows, and continuous dataflows. Flows can also be split or merged within a diagram.

Each dataflow item created is automatically entered into the data dictionary with no definitions. You must complete the definitions, which follow the DeMarco conventions, a Backus-Naur-form-like syntax with extensions. To complete a definition, you move the mouse cursor to the data name, press and hold the mouse button, and select Define in the menu. A data-dictionary entry box is brought to the screen with four fields: Name (filled in from the diagram), Definition, Description, and Domain. The data definition is BNF. Anyone familiar with this notation will be comfortable with the implementation. BNF definitions can be expanded to primitive data types such as integer or Boolean values. The description provides for explicit textual documentation for the entry. The domain specifies the legal values for data.

You can check your analysis for consistency and completeness in several ways. A check of the DFDs or data dictionary can be initiated from the main menu bar. A query window is brought up to report the results. Checks can also be done from within a diagram. These checks compare a diagram with its parent, its child, or both. The results appear in a query window, where you can click on the name of the offending item and issue a Goto command. TurboCASE takes you to the offending part of the analysis, either a diagram or a data item.

The decomposition is complete when further refinement does not assist in understanding the problem. Process modeling as either a minispecification or a decision table is provided at this point. Minispecifications are textual descriptions of the basic processes. Requirement traces can be set within the minispecification to facilitate traceability through the design.

Entity-relation modeling tools. The package also supports data modeling using Chen's notation. You can draw the diagrams, define entities and relationships with attributes, define the attributes, and check consistency of the data model or cross-check it with a functional model. The process of building entity-relation models follows the same techniques as the functional modeling.

Data modeling lets you perform object-oriented analysis that supports the Shlaer/Mellor methodology. TurboCASE does this by providing a link from each entity in an entity-relation diagram to a state-transition diagram. To create this link, you place the mouse cursor on the entity (object) and select the Life Cycle command. An ESTD (Entity State Transition Diagram) is created in which states can be defined and transitions or methods can be added to the definition.

Structured design tools. These tools let you create structure charts, define module specifications, and define the interfaces between modules. Developing structure charts is similar to developing a DFD. They are built with the usual caller/callee notation. The package also implements looping, case-switching, and a hierarchy of structure charts. You can expand any leaf module in a chart into a child chart. In this manner, the detailed parts of the design can be broken down in a logical and understandable manner.

The true beauty of TurboCASE's structured design lies in its procedures for defining modules and interfaces. All intermodule communications can be established from within the program. The module parameters and interfaces are defined in a consistent manner, with a checking mechanism for all module calls. Defining a module is accomplished by using a table containing the specification of its formal parameters. There are four parts for each parameter: name, type, usage, and scope. The parameter name and type can be any textual name and type for the planned implementation language. The usage can be input, output, input/output, or return. The scope is either local or global. Module-interface definitions are defined in a similar manner.

Designs are checked to see whether modules are specified correctly and fully defined in only one place. When modules and module interfaces are defined, TurboCASE checks design consistency in terms of the following rules:

- interfaces must be completely defined,
- the parameter lists for the two definitions must be the same length,
- the types and order of the parameters must be consistent,
- usages must be the same,
- functional usage must be consistent in the module interface,
- scopes need to be specified in the interface, and
- the parameter-passing of the child module must be defined.

Object-oriented design tools. The latest features added to version 4.0 deal with object-oriented design. The development of object designs and programming can be more complicated than traditional structured methods. Whether this complexity is inher-

ent to OOD/OOP or whether it is due to the unfamiliarity of programmers with this development paradigm, CASE tools can provide a smoother transition into object-oriented development. TurboCASE supports the concepts of several object-oriented methodologies and flexibly adapts to your particular methods. It provides four graphics editors and one dictionary editor for object-oriented development: Class Hierarchy Editor, Class Definition Editor, Class Collaboration Editor, Class Design Editor, and an OOD data-dictionary manager.

To create a Class Hierarchy diagram, choose Class Hierarchy under the modeling menu. You will be asked to give the diagram a name. You can now create classes of objects. You can show inheritance from one object to another or to a cluster of objects using the Inherit From, Inherit To, and Subclass commands. A connection line shows the hierarchy by placing a bar on the line close to the superclass object. You can also connect from a subclass object to a superclass object by using the Superclass command. To do either, choose Inherit From or Inherit To, move the cursor to the object that you want connected, and click on it. The connection is now made (as shown in Figure 4).

A class is defined through the Define command. It lets you define the class with another diagram or with a textual data-dictionary definition. The data fields in the data-dictionary block require you to identify the class name; any superclass or subclass; the purpose of the class; properties of the class; and abstractions, responsibilities, methods, and variables within the class. The name field is filled in when the window opens. The main menu in the data-dictionary window includes commands to Search, Add Control, Add Variable, and Add Method for the class object. You can also define a class object graphically. When defining the object, TurboCASE includes the commands Capability, Inst.Variable, Override Method, Abstract Method, Method, Impl.Abstract, Collaboration, and Define, as well as the usual drawing commands. Instance variables are displayed in a hexagon inside the class symbol. Methods are displayed in a rectangle, with abstract methods having a dashed line. Symbols of variables and methods are bound by the class, and you cannot move them out of the class symbol.

Multiple inheritance is supported because one or more subclasses can inherit methods from objects. A checking mechanism verifies that polymorphism works for your class hierarchy.

Class Collaboration modeling describes how object classes communicate with each other and how they coexist to accomplish a software task. The Class Collaboration editor is opened from the Modeling command in the main menu bar. Drawing is conducted as in the class hierarchy diagrams. As with all other TurboCASE tools, the OOD editors are integrated with the data dictionary. For instance, as you work in the Class Collaboration diagram editor, the correct information is entered into the data dictionary to maintain the consistency of your design.

TurboCASE also provides browsing similar to that provided in some OOP environments. If you correctly define a class hierarchy and the capabilities of the classes, the program shows you what that class provides. To view these, you open a graphical definition window for the object and select the Capabilities command from the menu in the object. From a dialogue window, you identify the superclass you are interested in. TurboCASE displays a query window detailing the capabilities of this class object and the source of their inheritance.

The final part of the OOD tool is the Detailed Class Logic Design. You can use the Design Model to diagram the design, define method signatures, and describe message-passing. Finally, the program checks your design, identifying inconsistencies and incomplete sections.

Summary

We are impressed with TurboCASE's quality and performance. The exceptionally user-friendly environment and consistent implementation enhance productivity; the check-and-query tools improve development quality. In addition to the analysis and design tools provided by TurboCASE, there are several options available for importing/exporting data to/from the TurboCASE database (which contains all information for a specific project). For instance, you can import/export an active window, a subpart of a functional project, the data dictionary, or the design models. You can also import/export to/from Cadre's Teamwork with an additional tool, the Teamwork data bridge.

You can also export Pict files from TurboCASE into other Mac documents. For instance, a drawing can be loaded and scaled in SuperPaint and pasted into an MS Word document.

The only weak spot is the user documentation. Although the manual concepts are good, some organizational changes would greatly improve the document. The manual provides two basic functions: tutorial and user reference. We wish that the tutorial were separated and completely developed. It would be very useful to the new user to have at least one complete development project example to follow, for both the functional structured methodologies and the object-oriented methods. In addition, there could be short demonstrations of each methodology tool. Completely missing from the documentation is a command reference section. Further, separating the user guide and commands from the tutorial would allow quick access for more experienced users.

Our testing has convinced us that TurboCASE is an exceptional CASE tool for software development. At $995, it can be used for analysis, design, and documentation development for a complete project or for any part of the effort. We believe that it could be used equally well by individual developers or teams through the use of the subproject import/export facilities. Further, because the minimum equipment necessary is a Mac Classic, the base installation cost is reasonable.

Readers can contact StructSoft at 5416 156th Ave. SE, Bellevue, WA 98006; (206) 644-9834; or circle Reader Service Number 21.

The power of a 486

— T.L. (Frank) Pappas, Aweb Software Technology

In this review I'll look at two ways to get the power of a 486 PC. I'll take a brief look at a really nice 486 that I found and a more in-depth look at 486 emulation on a Sun Sparc workstation.

A powerful 33-MHz 486

At $1,875, Comtrade's 33-MHz 486 midtower comes fully loaded with 4 Mbytes of 70-ns RAM and a 256-Kbyte cache; a built-in 487 math co-

processor; 1.2- and 1.44-Mbyte floppy drives; a 130-Mbyte IDE hard drive with a 64-Kbyte cache; a 16-bit 1,024 × 768 Super VGA card, 1 Mbyte of video RAM, and a 14-inch noninterlaced SVGA monitor; two serial ports, one parallel port, and one game port; a 101-key keyboard and a 400-dpi mouse; and both MS-DOS 5.0 and Microsoft Windows 3.1. For $90 you can upgrade to a 32-bit graphics card, and for $380 you can upgrade to 8 Mbytes of RAM and a 210-Mbyte hard drive. For those of you who need more horsepower, Comtrade also has a 50-MHz model.

The unit I looked at had the 32-bit graphics upgrade and 16 Mbytes of RAM. Since I planned on using an Orca SCSI drive in the review, I told Comtrade not to bother sending a hard drive.

This unit is amazing. Norton benchmarks showed a CPU speed index of 72.0 (more than twice that of the 33-MHz Compaq 386), a disk speed index of 9.3, and an overall performance index of 51.1. When I turned off the 256-Kbyte cache, I still got a CPU speed index of 54.6.

Within the system unit, there is plenty of room to hold a 32-bit slot (used for the graphics adapter) and seven 16-bit slots. Even the SIMM memory banks are easy to get to. The unit can hold two full-height hard drives. The top drive bay holds the 5.25-inch floppy and can accommodate a large disk like the Orca, but the second bay is designed to hold a 3.5-inch floppy and can only contain newer, trimmer full-height drives.

Comtrade uses an American Megatrends BIOS. When you boot the system, you can call up a collection of configuration menus for just about any aspect of the system you want. One option lets you define a password for system entry.

The three-button mouse can be configured as a Microsoft two-button mouse or as a three-button Mouse System mouse. There's also a wide range of video drivers for various programs such as Windows 3.0 and 3.1. When I used the 1,024 × 768 SVGA card with the 256-color driver in Windows, I was really impressed with the quality of the text and graphics produced by both the 32-bit graphics adapter and the SVGA monitor. It is well worth the $90 upgrade.

I ran DOS and SCO Unix on this unit and was extremely pleased with the performance. The only minor problem I found was the use of IRQ 2 for cascade-interrupt control of IRQ's 8-15, which prevents IRQ 2 from being used for peripherals as in most other 386/486 computers. For most people this won't be a problem, but it was for me. The floppy controller, the two serial ports, and the one parallel port each take an IRQ in the range 2-7. I also needed to install a Mountain 7000 tape unit and a 3COM 501 Ethernet card, both of which need IRQs in this range. Without IRQ 2, I was one short.

If I wanted to run DOS or Unix, but not both, I could get around this by using a DigiBoard PC/4 multiport card, which provides four serial ports but only uses one IRQ. In fact, I was able to do so for DOS and Unix. But since I wanted to use a mouse and an external modem on two of the ports, and use them under DOS *and* Unix, the PC/4 was no help.

As a reviewer, the computer I use must work well with any software, especially programs that use extended memory, so I tried just about every piece of DOS software that I had. This included Windows 3.0, Excel, Word, WordPerfect, Corel Draw, ZorTech C++, Borland C++, Alsys 286 Ada, Alsys 386 Ada, Meridian Ada, Tex, Norton Utilities, and much more.

If you are looking for a powerful, inexpensive 486 computer, and are not hung up on buying name brands, take a good look at this unit. Since prices are always changing, you might find a better deal — but I doubt it.

Readers can contact Comtrade at 1016-B Lawson St., City of Industry, CA 91748; (800) 969-2123; or (818) 964-6688; or circle Reader Service Number 22.

486 DOS on a Sparc workstation

As those of you who read this column regularly might know, I have been transitioning from DOS to Unix over the past few years. But like many Unix developers, I can't completely break away from DOS. In my case, it's because I develop both DOS and Unix software, and some of my clients require documents in Microsoft Word format. I also like to use Microsoft Excel spreadsheets and Corel Draw graphics. Finally, regardless of whether I am developing Ada or C++ software, whenever possible I compile the software under DOS *and* Unix to help ensure its portability.

Having ties to both DOS and Unix means that I have a 486 on my desk that runs Unix and MS-DOS, as well as a Sun IPC Sparc workstation. That takes up a great deal of valuable desktop space and also creates a web of keyboards, mice, and mouse pads. I have worked in the other extreme, where one of the computers I used was in another part of the building; I used to take manuals, tablets, a coffee mug, etc., to go work on something for a few hours. While the latter extreme is worse, the former isn't a great deal of fun either.

I'm sure that many of you are in the same situation. You may not cross-develop software, but if your company's standard word processor is something like Word, when crunch time comes during proposal-writing, you'll need to access a PC.

Some PC products are migrating to Unix. For example, Corel Draw for Unix, which runs like a typical X application, has the same look and feel as the Microsoft Windows version, but users of Micrographx Designer are not so lucky: a Unix version does not exist. Many DOS vendors seem to be ignoring Unix right now, but they may change their minds when they realize how much money can be made by migrating.

SunSelect, a Sun MicroSystems company, makes many of these problems disappear with its SunPC software and 486 accelerator cards. Unlike the DOS emulation products available for 386 and 486 Unix users, Sparc users can install a 486 accelerator card in a workstation Sbus slot and run 486 DOS applications in an X Window.

SunSelect offers three options. For $695, you can buy the SunPC software without the accelerator board to execute DOS programs that at most require an 80286. I think most users will find this option much too slow. A better choice is to get an accelerator card. The SX contains a 16-MHz 486 SX chip and lists for $1,495; the DX, which is the unit I reviewed, lists for

Benchmark test

I used the Norton Utilities SysInfo command to determine the performance of the SunPC Accelerator DX and the Comtrade 486/33. This command has a performance benchmark that reports a computer's CPU speed index, disk speed index, and overall performance index relative to a 4.77-MHz IBM XT (which always has an index of 1). For a Compaq 33-MHz 386, these values are 34.7, 8.4, and 25.9, respectively. — *T. Frank Pappas*

$1,995 and sports a 25-MHz 486 DX chip. The price of each includes the SunPC software (you need to run OpenWindows 3.0 to use it).

The installation was simple. I first installed the software, then the board. There are no jumpers or switches to deal with. I just opened the system unit and placed the card in the Sbus. There are no screws to fasten to the board as there are in a PC.

Although the SunPC software comes with MS-DOS 4.01 because of an existing licensing agreement with Microsoft, I easily upgraded to 5.0. (SunSelect will switch to 5.0 as soon as the agreement can be modified.) Instead of using a set of upgrade disks, I created a set of 3.5-inch bootable disks to install 5.0. I did that for two reasons. One was that my upgrade disks are 5.25 inches and the workstation has a 3.5-inch drive. I also thought this would be a better test of the emulation.

When you start SunPC for the first time, it creates a 32-Mbyte MS-DOS 4.01 c: drive (a Unix file). To upgrade, I placed my bootable 3.5-inch boot floppy into the workstation floppy (drive a:), entered CTRL-ALT-DEL to reboot, and formatted the c: drive with the /s option to install the new system files and command.com file. The lone difference I noted in the process was that it only took 2 seconds to "format" the c: drive.

This is a true DOS drive in the sense that only DOS commands can work on it. I could have created a second true 32-Mbyte DOS drive, but I chose to use an extended drive (a Unix directory) instead because I can use Unix commands on it. To make it available to a SunPC session, I simply had to add the line

```
net use d: /usr/local/dosapps
```

to my autoexec.bat file. You can have as many of these extended drives as you want. When SunPC is installed, the autoexec.bat has extended drives for your home and root directories.

To further test the emulation, I installed Microsoft Windows 3.1, using a SunPC-supplied SVGA driver, Excel 4.0, Word 2.0, the CD-ROM version of Corel Draw 2.0, Alsys V5.1 386 Ada, and Norton Utilities 5.0. All these programs worked exactly as I expected them to.

The workstation I used has a CD player, since Sun prefers to distribute software that way, so I asked Corel Systems for its CD version. I installed it under Windows 3.1. Once the installation was completed, I accessed some of the clip art from the CD. All this worked without a hitch.

I chose the Alsys compiler because it uses the PharLap 386 DOS extender and I wanted to see how it worked. I also knew that on a few computers, this extender has a problem. I compiled several Ada programs, including one that always reproduced the problem for me, but I had no problems on the DX.

As for the Norton utilities, the commands worked as expected. For example, I erased a directory and some files on the c: drive and on an extended drive. I could unerase the files on the c: drive, but the Unerase command could not detect any erased files or directories on the extended drive. When I ran the Norton benchmarks (see the sidebar on p. 88), they reported a CPU speed index of 56.8, which seems reasonable; but the disk speed index gave nonsense results (which I expected).

For the most part, SunPC is well integrated into the OpenWindows environment. For example, you can cut and paste between the SunPC window and other windows. Although the SunPC window is a fixed size, SunPC does come with an SVGA driver to use when installing MS Windows. When you start MS Windows, the SunPC window increases to give a better Windows desktop.

The left and middle buttons of the workstation mouse act as the left and right buttons of a two-button Microsoft mouse. The third button pops up a menu of useful services, one of which is to attach and detach the mouse to the SunPC window. When attached, the mouse acts as a typical Microsoft mouse, but it cannot be used to select other X windows. You must detach the mouse if you want to use it in other windows. Another menu option ejects the floppy from the a: drive. This is needed because a workstation does not have a floppy disk eject button.

A good feature of the SunPC software is that it provides keyboard shortcuts for the choices on the pop-up menu. For example, META-e ejects the floppy, META-m attaches and detaches the mouse, and META-r reboots the SunPC window.

The only awkward feature that I found was the need to attach and detach the mouse. The SunPC project manager told me SunSelect feels the same way and is trying to come up with a better way of sharing the mouse. Until that time, this minor awkwardness is a very small price to pay for the power and usefulness of the SunPC emulation.

There are many more emulation features that I don't have the space to get into, but overall I strongly recommend the 33-MHz 486 Sun Accelerator DX. The SX version, while slower, is probably still a good second choice, but I think that just the SunPC software alone would be a disappointment.

Readers can contact SunSelect through HiTech Public Relations, 101 Howard St., Ste. 200, San Francisco, CA 94105-1616; or circle Reader Service Number 23.

Object-oriented programming in Ada

— T.L. (Frank) Pappas, Aweb Software Technology

Those of you who program in Ada know that although you can write object-oriented Ada programs, this language doesn't offer the inheritance and polymorphism available in OOP languages like C++ or SmallTalk 80. The ANSI Ada revision, referred to as Ada 9X, will correct much of this. Until Ada 9X is available, programmers can use Classic-Ada from Meridian Software Systems. Classic-Ada is a SmallTalk 80-like OOP front-end translator for Ada, just like AT&T's cfront is a C++ front end for C.

I reviewed Classic-Ada a few years back when it was available only for Unix systems, so I won't do a full review on it now, but I thought a short update on the PC version might interest some readers. Figures 1a and 1b on the next page give a flavor of OOP in Classic-Ada. Figure 1a is an example of a Classic-Ada class declaration.

Like an Ada package, a class declaration consists of both a specification and a body, which can be compiled separately. Classic-Ada maintains a class library similar to Ada's program library to ensure consistency between the specification and body parts as well as the use of the specification.

The Classic-Ada compiler translates a class into an Ada package, checking the syntax and semantics of the Classic-Ada features but only checking the syntax of the Ada features actually used. The resulting translation must be compiled by an Ada compiler. A supplied Classic-Ada executive must also be compiled.

An interesting aspect of Classic-Ada is that it is compiler independent; that is, the code generated by the program can be compiled by any Ada compiler. You can buy Meridian's Classic-Ada and use it to develop OOP programs under, say, Alsys Ada. However, if you plan to use Classic-Ada without the company's 32-bit compiler, make sure you let Meridian know. The program requires an executable file, OS386.EXE, that is shipped with the 32-bit compiler; but since Meridian has to pay a royalty for this executable, it is not normally shipped with the package.

I think Classic-Ada, which sells for $349, is a viable way to get into OOP with Ada. But you should be aware that the OOP extensions being added to Ada 9X extend Ada packages but do not add a class construct. However, it should not be difficult to transition Classic-Ada programs to Ada 9X.

The add-on package costs $349. Readers can contact Meridian Software Systems at 10 Pasteur St., Irvine, CA 92718; (714) 727-0700; or circle Reader Service Number 24.

```
WITH String30;
CLASS Object1 IS
    METHOD Create (New_Instanee : IN OUT Object_Id);
    INSTANCE METHOD Initialize_Message;
    INSTANCE METHOD Initialize;
    INSTANCE METHOD Say (Greeting : IN String30.String_Type);
    INSTANCE METHOD Finalize;
    INSTANCE METHOD Delete;
END Object1;

WITH Text_IO;
CLASS BODY Objeet1 IS
    Instanee_Message : INSTANCE String30.String_Type;

    METHOD Create (New_Instanee : IN OUT Objeet_Id) IS
    BEGIN
        New_Instanee := INSTANTIATE;
    END Create;

    INSTANCE METHOD Initialize_Message IS . . .

    INSTANCE METHOD Initialize IS
    BEGIN
        Text_IO.Put_Line (String30.String_Of (Instanee_Message) );
    END Initialize;

    INSTANCE METHOD Say (Greeting : IN String30.String_Type) IS . . .

    INSTANCE METHOD Finalize IS
    BEGIN
        NULL;
    END Finalize;

    INSTANCE METHOD Delete IS
    BEGIN - - SELF and DESTROY are Classic-Ada reserved words.
        Send (SELF, Finalize);
        DESTROY;
    END Delete;
END ObjeetI;
```

(a)

```
CLASS Objeet2 IS
    SUPERCLASS Object1;
    METHOD Create (New_Instanee : IN OUT Object_Id);
    INSTANCE METHOD Initialize;
    INSTANCE METHOD Finalize;
    INSTANCE METHOD Delete;
END Objeet2;

WITH String30;
CLASS BODY Objeet2 IS
    METHOD Create (New_Instance : In OUT Object_Id) IS
    BEGIN
        New_Instanee := INSTANTIATE;
        SEND (New_Instanee, Initialize);
    END Create;

    INSTANCE METHOD Initialize IS
    BEGIN
        String30.Assign (Instanee_Message, " Initialized 2");
        SEND (SUPER, Initialize);
    END Initialize;

    INSTANCE METHOD Finalize IS
    BEGIN
        Send (SUPER, Finalize);
    END Finalize;

    INSTANCE METHOD Delete IS
    BEGIN
        Send (SELF, Finalize);
        SEND (SUPER, Delete);
        DESTROY;
    END Delete;
END Objeet2;
```

(b)

Figure 1. Example of a Classic-Ada class declaration (a) and a second class descendant from the declaration (b).

CHAIR, ELECTRICAL ENGINEERING

The Duke University School of Engineering invites applications for the position of Chair of the Department of Electrical Engineering. The Chair is responsible for the faculty, teaching programs, and research activities of the department, and reports to the Dean of the School. Eligibility for appointment as a full professor at Duke, with the academic credentials thereby required, is essential.

Interested persons should write to the **Search Committee for the EE Chair, Office of the Dean of the School of Engineering, 305 Teer Bldg., Duke University, Durham, NC 27706, including a complete curriculum vita and the names of four references. Applications should be received by August 1, 1992.**

Duke University

Duke University Is An Equal Opportunity/Affirmative Action Employer.

IC Announcements

Company, Model, Function	Comments	R.S. No.
ACC Microelectronics ACC2168 Microcontroller	Integrates control logic in a single 208-pin package for 386DX or 486 PC/AT 32-bit designs. Accesses up to 64 Mbytes of on-board system memory while supporting up to 2 Mbytes of direct-mapped cache. Includes a local bus interface, a standard AT bus interface that supports synchronous or asynchronous clock timing, and an integrated peripheral controller. Available in frequencies from 16 to 50 MHz. Price (in 1,000s): from $35.	121
AMI 18CV8-7, 22CV10-10 PEEL devices	Programmable electrical erasable logic devices offer 7.5-ns operation in the 20-pin 18CV8-7 configuration and 10-ns operation in the 24-pin 22CV10-10 configuration. Fabricated with CMOS EEPROM technology to interface with RISC-type microprocessors while consuming 90 mA of power. A variety of packaging options will be available beginning third quarter 1992. Price (in 1,000s): $8 (18CV8-7); $11 (22CV10-10).	122
Anadigics Cable TV converters GaAs chip	Designed to Jerrold Communications specifications, this IC provides front-end functions for double-conversion, 84-channel cable TV converters — replacing more than 30 discrete components, according to company sources. Functions include mixer, oscillator, RF amplifier, and automatic gain control. Comes in a plastic DIP.	123
ATI Technologies 68800 Graphics chip	Single chip incorporates graphics acceleration, linear memory aperture, and VGA. Supports EISA, ISA, and Micro Channel buses, as well as VESA's Local Bus design specification for a direct 32-bit link between the CPU and peripheral devices. Price (in 1,000s): $79.	124
Cyrix Cx486DLC Microprocessor	This 486-instruction-set-compatible CPU with 32-bit internal and external data paths is available with 25-, 33-, and 40-MHz clock speeds. Incorporates a single-cycle pipelined processor core with a 1-Kbyte data and instruction cache. Available in a 132-lead 386DX-compatible footprint. Price: $119 (33 MHz).	125
Linear Technology LTC1096 A/D converter	Converter draws less than 100 µA of supply current when sampling at its maximum rate of 33 kHz and falls to 300 nA at 100 Hz. Operates from a single 3V-to-9V supply and automatically powers down to a supply current of 1 nA (typically) between conversions. Packaged in 8-pin plastic DIPs and SO-8 packages. Price (in 100s): $3.95 (DIP), $4.25 (SO-8).	126
Mitsubishi M6007X, M6008XL Gate arrays	Low gate-count versions of 0.8-µm array families include M6007X series (6,000 to 20,000 gates) and M6008XL (25,000 to 50,000 gates). Four speed/power options attain a typical loaded delay as low as 215 ps and power dissipation as low as 2.4 µW/MHz/gate at 5V. Available in wire bond QFPs with up to 208 pins. Designs are being taken for QFP and PGA packages.	127
Motorola 68HC11 family Microcontrollers	The 8-bit 68HC11 microcontroller family has been extended to include five low-voltage (3V to 5.5V) versions: A8, E9, D3, L6, and K4. According to company sources, the new versions are suited for applications requiring a longer battery life, such as hand-held and portable systems. Various packaging. Prices (in 10,000s): from $7.94 (A8) to $15.86 (K4).	128
Nat'l Semiconductor DP83955, DP838956 Interface controllers	Lite Repeater Interface Controllers provide six ports with integrated 10BASE-T transceivers plus one AUI port for connection to all 802.3 media. The DP83955 is designed for 802.3 concentrators that support departmental LANs of up to 200 nodes; the DP83956, for concentrators that require departmental and enterprise-type hub management. Sample quantities available. Price (in 1,000s): $37 (DP83955), $43 (DP83956).	129
Philips Semiconductors 74HL33240 Octal buffer line driver	Three-state device is the first of approximately 20 logic ICs proposed for release this year in the "HLL" family. According to company sources, all HLL devices operate on 3.3V power supplies while running twice as fast as bipolar 5V logic circuits and consuming half the power of equivalent 5V CMOS types. Typical propagation delay for the 74HL33240 is 2.5 ns.	130
Trident VideoView Video chip set	Two-chip set for scalable full-motion video windowing using Microsoft's Windows or DOS: T1 offers the bus interface, memory control, and display control; T2 consists of the live-video processor, digital special effects, overlay control, and VGA interface. Price (in 100s): $120.	131
VLSI Technology VL82C380 Memory controller	PC/AT-compatible 32-bit single-chip controller with on-chip cache is designed for use in 386DX-based ISA systems operating at clock speeds up to 40 MHz. Its cache controller has a "look-aside" write-back architecture. Sample quantities available in a single plastic 208-lead metric QFP. Price (OEM quantities): $20.	132

Microsystem Announcements

Company, Model, Function	Comments	R.S. No.
Accton Technology EN1641/2, EN1843/4 Network interface cards	Accton's EN50903 Ethernet controller chip is the basis of four adapter cards: EtherPair-16/EtherCoax-16, EtherPair-8/EtherCoax-8. EtherPair cards are for unshielded-twisted-pair networks through an AUI or RJ-45 connector. EtherCoax cards are for coaxial cable networks with software-selectable settings for an AUI or BNC adapter. All four NICs offer up to 256 software-selectable I/O addresses.	141
Alta Technology HSI-SBus Host interface card	Host System Interface for the SBus is a one-slot card for connecting Sparc-based workstations to scalable compute engines and peripheral devices. Incorporates a proprietary 32-bit interface between the SBus and a 25-MHz T805 transputer. Provides sustained 10-Mbyte/s transfer between SBus and transputer. Price: from $1,595.	142
Alta Technology SCSI TRAM SCSI-2 interface	Size-4 transputer module incorporates a proprietary 32-bit interface between NCR's 53C710 SCSI-2 processor and a 30-MHz T805 transputer. Permits attachment to existing SCSI-1 or -2 controllers or devices. Provides sustained 10-Mbyte/s transfer rate on the SCSI-2 bus that is matched by either a 25- or 30-MHz transputer and four Mbytes of zero-wait-state cache memory. Operates in either target or initiator mode. Price: from $2,295.	143
Ampro Computers Little Board/386SX Single-board computer	SBC offers standard 25-MHz 386SX AT motherboard features, including up to 16 Mbytes of onboard DRAM, in a 5.75×8-inch embedded SBC that requires 5W of 5V-only power. Other onboard features include two serial ports, one parallel port, a SCSI interface, and a watchdog timer. Operates at +5V. Price (in 100s): $720.	144
Communication Automation & Control VME6U6 6U VMEbus board	Board includes six 50-MHz 32-bit floating-point AT&T DSPs and features peak performance of 150 Mflops and 75 MIPS. Each DSP executes programs and accesses data from a local 512-Kbyte zero-wait-state SRAM. Board can act as a VMEbus master or slave. Optional daughter cards available for T1 interface and Codecs. Price: $10,900.	145
Heurikon HKMIPS/V3000 VMEbus CPU boards	Based on the 25-MHz Mips R3000 RISC processor, the board features a 25-MHz R3010 floating-point coprocessor, 12 Kbytes each of direct-mapped data and instruction cache, up to 16 Mbytes of local DRAM, 512 Kbytes of boot PROM, and 128 Kbytes of EEPROM. VMEbus interface is a full 32-bit Slot One master/slave interface. Price: from $6,995.	146
Iomega PC Powered 90, PC Powered 90 PRO Bernoulli drives	Two 90-Mbyte Bernoulli drives measure 2.75×6.75×10.75 inches. They operate with standard AT (ISA) bus PCs and receive power from the computer through the bus adapter card included with the unit. The PC Powered 90 PRO uses Iomega's PRO drive for 15% faster performance and provides SCSI capability. Price: $713 (90), $855 (90 PRO).	147
PEP Modular Computers VM-30 SBC	Single-height (3U) dual-processor SBC with 8- or 10-MIPS performance consumes 5.5W of power. Host processor is a 25- or 40-MHz 68EC030 chip that handles I/O and processing for compute-intensive and number-crunching applications. The second processor is a 16.7-MHz 68302 intelligent multiprotocol processor that handles I/O for synchronous and asynchronous data communication protocols. Price: $1,995.	148
Radstone Technology FDDI-1 VMEbus controller	Intelligent controller provides a fully compliant dual- or single-attach FDDI node in a single VMEbus slot. Features include a SparcLite embedded processor, separate program and data memory segments, and tag mode buffer memory management. Supports concurrent VME transfers, protocol processing, and 100-Mbit/s data transfers. Price: from $9,500.	149
Sealevel Systems Comm+232/EX Serial interface card	Lets users of XT/AT compatibles add COM1 through COM4 to a Windows 3.1 system, using either shared interrupts or extended AT interrupts (IRQ 10, 11, 12, and 15). Includes two independent RS-232 serial ports, selectable port addresses (COM1 through COMX), all RS-232 modem signals, connectors, and utilities that control the ports on its board. Price: $179.	150
Texas Instruments PCMIA-standard Memory cards	OTP and flash memory cards based on the Personal Computer Memory Card International Association standards are available in 256-Kbyte, 512-Kbyte, and 1-Mbyte densities. Price: from $72 (OTP, 256 Kbytes) to $521 (flash, 2 Mbytes).	151
Voice Connexion IntroVoice Pro-Module PC voice system	Voice recognition and speech synthesis system has MiniModule form factor designed for compatibility with Ampro's CPU motherboards—CoreModules and LittleBoards. Provides voice recognition of 500 words with 98%+ accuracy and unlimited text-to-speech synthesis. Features DOS-compatible software for building vocabularies and training. Price: $495.	152

Computer Society NEWS

Editor: Guylaine M. Pollock, Sandia National Laboratories, Division 1424, PO Box 5800, Albuquerque, NM 87185; phone, (505) 845-7463; Internet, gmpollo@cs.sandia.gov

Slate of candidates approved for fall election

The IEEE Computer Society Board of Governors has approved the slate of candidates for president-elect, first vice president, second vice president, and seven board posts that will be filled by membership vote this fall. The board took action on the slate when it met in New Orleans June 5.

Those elected, including the seven candidates receiving the greatest number of votes among the 12 Board of Governors nominees, will begin their terms January 1, 1993.

The candidates are as follows:

1993 president-elect (1994 president, 1995 past president)
(One elected)
Ronald G. Hoelzeman
Laurel V. Kaleda

1993 first vice president
(One elected for one year)
Joseph Boykin
Gerald L. Engel

1993 second vice president
(One elected for one year)
Fiorenza C. Albert-Howard
Anneliese von Mayrhauser

1993-95 Board of Governors
(Seven elected for three years)
Hasan S. AlKhatib
Fletcher J. Buckley
Doris L. Carver
Elliot J. Chikofsky
David M. Choy
JoAnne E. DeGroat
Michael J. Flynn
Tadao Ichikawa
Mary Jane Irwin
Vincent Y. Shen
Grace C.N. Wei
Akihiko Yamada

Additional nominees can be included on the ballot by membership petition (see *Computer*, February 1992, pp. 83-84, for a complete list of Computer Society requirements for petition candidates).

Petition candidates must submit their petition signatures (1,000 voting members for officer nominees and 250 voting members for Board of Governors nominees) and their biographical data, photographs, and position statements to the Computer Society secretary at the following address on or before July 31, 1992:

Ronald G. Hoelzeman
University of Pittsburgh
Department of Electrical Engineering
348 Benedum Eng. Hall
Pittsburgh, PA 15261

Candidates' statements and biographical data will be published in the September 1992 issue of *Computer* and will be included in the August 14, 1992, IEEE ballot mailing. Length limitations for these materials are as follows:

Candidates' statements
President-elect — 350 words
Vice president — 250 words
Board of Governors — 150 words

Biographical data
All candidates — 200 words

Biographical sketches should cover the following topics in the sequence listed:

- Computer Society activities
- other professional activities
- current employment, professional experience, and accomplishments
- degree(s) and majors(s)
- awards and honors

Nominees should also submit black-and-white passport-type photographs of themselves for publication along with their statements and biographies to

Bob Carlson
Computer Magazine
10662 Los Vaqueros Cir.
PO Box 3014
Los Alamitos, CA 90720-1264
E-mail, b.carlson@compmail.com

Flynn receives 1992 Eckert-Mauchly Award

Michael J. Flynn, professor of electrical engineering at Stanford University, was presented the 1992 Eckert-Mauchly Award at the 19th International Symposium on Computer Architecture, held May 19-21 in Queensland, Australia. He was cited for "important and seminal contributions to processor organization and classification, computer arithmetic, and performance evaluation."

After joining IBM in 1955, Flynn worked for 10 years in computer organization and design. He was design manager of the System 360 Model 90 series. During 1973-74, he served as cofounder and vice president of Palyn Associates, a computer design firm in San Jose, California, where he is now a senior consultant. He joined the faculty of Stanford University in 1975 and was director of the Computer Systems Laboratory from 1977 to 1983.

Flynn founded Stanford's Computer Emulation Laboratory, which has been a leading facility for the analysis of computer architecture. His current research projects include programs on ultra high speed arithmetic performance, rapid evaluation of computer architectures, and parallel machines.

An IEEE Computer Society member, Flynn has served on the society's Board of Governors and as associate editor of *IEEE Transactions on Computers*. He was founding chair of both the ACM Special Interest Group on Computer Architecture and the Computer Society's Technical Committee on Computer Architecture.

The Eckert-Mauchly Award is presented jointly by the ACM and the Computer Society each year for outstanding contributions to the field of computer architecture.

Opening doors in the global village

Ross Alan Stapleton
US Central Intelligence Agency

(This material has been reviewed by the CIA to assist the author in eliminating classified information, if any; however, that review neither constitutes CIA authentication of material nor implies CIA endorsement of the author's views.)

With enactment of the High Performance Computer and Communications Initiative (HPCCI), the US is a step closer to the realization of a National Research and Education Network. The NREN will serve as an "Information Age interstate," and it is being sold first and foremost as a tool for advancing US technological (and thus economic) competitiveness. But this information superhighway is, in the words of the late Ithiel de Sola Pool, one of many "technologies without boundaries."[1] The concrete interstate highway system, for all it has enabled, still constrains us to follow the earth's geography: We are physically confined to a small portion of our hemisphere. But with an electronic superhighway, no place on the globe need be more than a few seconds away.

The US has the world's largest information economy, though other nations (particularly several neighbors across the Pacific) may be surpassing the US in the degree to which information forms the basis of national wealth. It is time for the nation to address both the formation of a domestic information infrastructure and the updating of its foreign technology policy. The former will need to be competitive with those of foreign counterparts, and the latter should reflect the new realities that the information technologies are creating.

Building an information state. The IEEE membership should be keenly interested in the growth of the domestic networks — the NREN as well as the many information services that are not part of this network. The NREN is intended as a possible model for commercial networking, though it has grown spectacularly and somewhat anarchically through sizable contributions of community resources. Simultaneously, enormous changes are likely in commercial information services, as suggested by negotiations over the role telephone companies will play in the creation and collection of information. The relationship and evolution of these two enterprises will shape the resulting Information Age landscape and dramatically affect how we work and live in this new era.

The US is not alone in facing these challenges, and it would do well to observe experiments elsewhere. France, for example, has made networked information broadly accessible through Minitel, enabled by the monopoly power of the state telephone service. While the US enjoys the benefits of competition between communications carriers, providers have repeatedly failed to deliver anything resembling Minitel's videotex-based, value-added services. The majority of French telephone subscribers have a national electronic telephone directory at their fingertips, along with thousands of other services; Americans face an electronic landscape dotted with oases but still largely desert for vast stretches.

When the US built its interstate highways, it had a great many more "rules of the road" than exist today for the information interstates. It could be a mistake to overestimate the virtues of unchecked diversity. While it may not be wise to put complete faith in standards, the networks deserve more attention as a national resource than they have thus far received. As they increasingly form the infrastructure for the US economy, they'll need to be made more coherent — and transparent. The question for the next several years is how to best put information resources into play in such a way that the content, and not the carrier, is all we care about.

The networks have helped to prove the viability of the information economy, contributing across the board in areas from scientific research to education; but they cannot be expected to function so smoothly as they change from an experimental luxury to a necessity. Their growth will require the equivalent of a highway planner. The first incarnation of the HPCCI died in Congress in 1990 amidst interagency squabbling. The initiative has now been enacted, but there is still work to do in the formation of a domestic information traffic policy.

Recognizing the arrival of the global village: Living with our neighbors. While considerable interest and attention have been devoted to the formation of a national network linking high-performance computing within the US, there has also been anxiety in some quarters about the spread of such technologies to other countries. Moreover, with the end of the Cold War, there has been a casting about for new problems to fit old answers of export controls. Because of its dual nature, networking may very well be the straw that breaks the back of past policies: It is simultaneously a technology in its own right as well as the means to make far more of other technologies.

As a technology, networking is subject to export controls. Because the computers and communications hardware and software necessary for high-speed networking are capable of "dual-use" — both civilian and military applications — they are regarded as resources that should be denied to hostile nations. Prior to the Soviet collapse, such nations were more easily defined, and the US, with CoCom (Coordinating Committee for Multilateral Export Controls), enforced a de facto embargo on a sizable share of information technologies.[2] In the post-Cold War era, there is a far more diverse neighborhood of nations, and it is harder to determine who may pose a threat.

Networking, however, has the power to turn the concept of export controls virtually inside out. There are few if any mechanisms to enact effective export controls on information, and according to existing legislation, there is no justification for trying to control information "in the public domain" — for example, the original work churning out of research departments in academia, industry, and government to be widely shared in the US tradition of academic openness.

Donald Douglas, cofounder of McDonnell Douglas, once said, "When the weight of the paperwork equals the weight of the plane, the plane will fly"; the same might be said of the pa-

The society's Committee on Public Policy examines a wide range of issues and periodically offers its findings in *Computer*. Ideas expressed under the COPP banner are not necessarily official positions of the IEEE Computer Society.

perwork required to install a supercomputer overseas. US and Japanese supercomputers are in use in a variety of countries, from Brazil to India. Each installation required advance assessments of the likelihood that the supercomputer, once acquired, might be diverted to illicit uses (such as designing bombs). Each transfer entailed the drafting of a security plan and arrangements for site inspections at agreed-upon intervals. It has been the CoCom community's business to watch such things very closely.

Now, however, networks have erased the physical borders that made such restrictions a practical policy. According to the *National Science Foundation Link Letter*, for example, scientists in Australia recently demonstrated the use of a Connection Machine located in Chicago, via the internetworking of Australian and US networks. Similar demonstrations could have been conducted as well from South Africa, Czechoslovakia, or Poland, all of which joined the global Internet within the last year. On the one hand we try to frustrate potential abusers of supercomputers overseas by attempting to meet all the licensing requirements, which imposes extra costs in the form of supercomputer sales delayed or never made. On the other hand we pay minimal attention to the many more powerful machines in the US that are becoming globally accessible. How can such a policy be reconciled?

There is a flow (some might consider it a potential hemorrhage) of information and computational resources out of the US. Traditionally, this gives rise to two major concerns: revelations that might diminish national security, and the uncompensated dissemination of US intellectual capital.

How do we place a value on information? And how do we calculate the costs and benefits of its control, if indeed it can be controlled? By recently becoming a signatory to the Berne Convention, the US has agreed to abide by broad international definitions of intellectual property rights; at the very same time, networks are making information more mobile than ever before. Practically speaking, there may be no way to police much of the global information transfer in the age of global information highways.

The problem is similarly complex when we try to weigh the costs and benefits of networking in terms of national security. One of the highlighted concerns in the "new world order" is the "proliferation of weapons of mass destruction" — the acquisition by less technologically advanced nations of the same class of military arms that the major powers have had for years. Networking, we can reasonably expect, will play a part in this proliferation. At a critical juncture, the US drew on the intellect of immigrant scientists (many of the contributors to the Manhattan Project were, in fact, early emigrants from a Europe sliding toward war). Now, any nation hoping to acquire its own capacity to produce modern weapons can comb the globe for assistance. In the 1990s, this need only entail the migration of information — of ideas — not of individual scientists.

Of course, there is a beneficial side to all this. Networks grant us the power to gather resources from all over the globe by allowing scientists and scholars to correspond with foreign counterparts. To a degree, such access to the world will defuse the very threats the outflow of certain information might fuel. Networks have the potential to lessen ignorance of foreign cultures — to help dispel myths and misunderstandings. If the US is to become an information sieve in this Information Age, it can also benefit as other countries become less opaque as well.

The global information landscape varies enormously. Outside of the industrialized nations, the pervasiveness of the information technologies falls off dramatically. This is not to say that the networks cannot give us access to significant portions of the developing world: We just have to recognize the appropriate technologies that provide the tools. In the former USSR, simple message-passing protocols, PCs, small minicomputers, and the switched tele-

ADVANCED SYSTEMS RESEARCH

The Institute for Defense Analyses (IDA) offers challenging career positions in its Computer and Software Engineering Division to experienced individuals with strong backgrounds in computer science. As a Federally Funded Research and Development Center, IDA addresses a wide range of complex national security issues for the Office of the Secretary of Defense, the Joint Chiefs of Staff, Defense agencies, and other Federal sponsors.

Current areas of activity in the Computer and Software Engineering Division include: software engineering, distributed systems, very large scale information management, system architecture, advanced decision support, simulation, and enterprise integration. While opportunities exist for individuals in all these areas, particular consideration will be given to candidates with technical and analytical experience in:

- **Distributed Computing**
- **Networks**
- **Computer Security**
- **Operating Systems**
- **Computer Architecture**

Doctoral-level knowledge and experience are highly desirable.

IDA offers excellent opportunities for professional development in a superior environment. You can expect a competitive salary, excellent benefits, and the chance to contribute substantively to the solution of urgent and critical problems facing our nation, while at the same time playing a role at the leading edge of the state of the art in computing. If this kind of future appeals to you, we urge you to investigate a career with IDA. Please forward your resume to:

Mr. Thomas J. Shirhall
Manager of Professional Staffing
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

An equal opportunity employer, m/f.
U.S. citizenship is required.

IDA

phone system have permitted the creation of a sprawling and fast-growing network — Relcom — that now links hundreds of computers and thousands of users from the Baltics to the Caucasus to Eastern Siberia.

The US and other industrialized nations will find their own interests served by keeping a finger on the pulse of global science and the technologies advancing in a hundred countries worldwide. Forty years ago the US was comparatively isolated, inventing the hydrogen bomb and the intercontinental ballistic missile on one side of the Atlantic while the Soviets did the same on the other side. The countries spied on each other, and much of the information that flowed between the two technological communities came through the diplomatic "networks" of embassies and consulates. Governments in and of themselves are, with some oversimplification, little changed in their capabilities. They have, however, begun to use a much greater variety of contacts and resources. An example of the rising importance of private versus government power in amassing information is CNN, the Cable News Network, which is arguably one of the most important information sources available to the US government.

But the considered, in-depth assessments required to ensure national security will require something more than CNN. There is little question that the government should maintain its human network of science and technology attaches and counselors with their direct access to developments in other nations. Each diplomatic position the US staffs abroad is expensive, however. To keep a single science attache in Moscow costs approximately $100,000 per year above and beyond the individual's salary, and much more in places like Tokyo. A leased communications line between the extensive US domestic networks and the Relcom network radiating from Moscow would be no more expensive than that solitary individual and could immediately pull two whole communities substantially closer together.

Of course, there is no reason for the government not to emphasize both, maintaining a physical presence abroad even while expanding a virtual presence through the networks. While compiling material for this article, I corresponded via electronic mail with David Kahaner of the Office of Naval Research, which maintains a liaison office staffed with scientific experts in Tokyo and a second office in London. Kahaner has access to the Internet through a Japanese university and is thus able to send back a stream of reporting on Japanese activity in supercomputing and artificial intelligence, as well as the various aspects of Japanese government programs — for example, the Real-World Computing Program (formerly New Information Processing Technologies), sponsored by MITI. This reporting is immediately available to a broad audience in the US and anywhere in the world the networks reach. Equally important, Kahaner is available as well.

The networks may also be the cheapest way to enable the sort of citizen diplomacy that President Bush called for in May 1990.[3] He spoke in the context of the vacuum left in Eastern Europe by the retreat of the USSR. The opportunities today, with the actual collapse of the Soviet Union, are even greater. Cynics might interpret the call for citizen diplomacy as a desire to pass the buck on foreign aid, but it is also apparent that the information technologies are empowering private individuals and groups as never before. Individuals and government could take up such a challenge, working together for their mutual benefit.

Even with the phenomenal progress in making computers faster, more powerful, and better able to augment the human mind, the most important resources on the networks are still people. Much of the "advertising literature" on the coming NREN and the rest of the HPCCI tends to stress only the highest end of the application spectrum: the transfer of complex medical scans and multidimensional dynamic models to large color workstations at megabits or gigabits per second. There is less emphasis on the networking of people. Perhaps this is because the organization of electronic communities will be accomplished not by any new technologies that the HPCCI might fund, but by an investment in time, community resources, and our own reeducation as we learn to perceive ourselves as members of the global electronic community.

In the past few years, the electronic mail address has become ubiquitous in some professional communities. Starting from zero just three or four years ago, calls for papers and conference announcements in *Computer* and the *Communications of the ACM* now list e-mail points of contact well over 80 percent of the time — nearly 100 percent for US events. But the e-mail address is far more a personal tag than an organizational tool: Within our electronic space we need to build the same sort of organizational structures we have in the traditional world.

Some recommendations. The US should define a modern information technology foreign policy in light of the new political and technological realities. Balancing the pros and cons of exporting networking technologies should include in the equation the very real gains that might be made by strengthening the bonds between the electronic communities and affording the nation better access to the rest of the world. The US government could play a role in building bridges, thereby enticing its citizens to take ideas and expertise abroad for the good of both the US and those countries that might be brought into closer collaboration. This will require the leveling of some remaining barriers, especially those whose raison d'etre has been made obsolete by the transformational power of information technologies. When we have cleared away the obstacles justified only by our Cold War fears, we ought to consider investing in the global infrastructure, though merely reducing bureaucratic barriers may be enough to entice private interests to make that effort.

For our part, as individuals and representatives of nongovernmental societies, we need to foster the extension of the professional communities along these new information roadways. By recognizing the new realities and taking advantage of the information technologies now in the hands of private citizens, we can help to fill the vacuums left in the wake of the enormous political changes that have recently swept the globe.

References

1. I. de Sola Pool, *Technologies Without Boundaries: On Telecommunications in a Global Age*, E.M. Noam, ed., Harvard University Press, Cambridge, Mass., 1990.

2. "Finding Common Ground: US Export Controls in a Changed Global Environment," Panel on the Future Design and Implementation of US National Security Export Controls, National Academy of Sciences, National Academy Press, Washington, D.C., 1991.

3. "Bush Announces 'Citizen Democracy Corps,' " UPI, May 12, 1990.

Ross Alan Stapleton, an analyst with the Central Intelligence Agency, has a doctorate in business and public administration. He can be contacted at the Central Intelligence Agency, Washington, DC 20505, or by e-mail at r.stapleton@compmail.com.

Send news about research, public policy, or professional issues to Bob Carlson, *Computer*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264

Caltech's Delta supercomputer studies galaxy formation

Ware Myers, Contributing Editor

By running large simulations on the Intel Touchstone Delta massively parallel supercomputer, located on the campus of the California Institute of Technology, two astrophysicists have made a significant step toward understanding the formation of the universe.

For this work, John Salmon of Caltech and Michael S. Warren of Los Alamos National Laboratory received the Grand Challenge Computing Award of $35,000. The winning team was selected by the Policy Board of the Concurrent Supercomputing Consortium, an organization of 14 universities, laboratories, and federal agencies for which Caltech operates the Delta. Intel's Supercomputer Systems Division provided the prize money.

The award was the high point of ceremonies on May 27 at Caltech celebrating the first anniversary of the supercomputer's installation (see *Computer*, July 1991, pp. 96-97). In addition to the winning team, a dozen other investigators displayed their efforts at a computational science fair. Still other teams reported highlights of their work on videotape.

The value of the research done during this first year was lauded in talks by Thomas E. Everhart, Caltech's president; Eugene Wong, associate director for industrial technology in the White House Office of Science and Technology Policy; and Stephen Squires, director of the Computing Systems Technology Office, Defense Advanced Research Projects Agency.

"The whole concept of this consortium and what it has done has clearly accelerated the maturation process of the technology," Squires said. He backed up his praise by announcing that DARPA has decided to increase its second-year support for the consortium from the planned $250,000 to $1 million. He urged other members who could afford it to increase their dues by similar proportions.

In a further expression of support for the consortium approach, Squires announced that "DARPA, with the National Science Foundation, is in the process of forming a national consortium for high-performance computing. We expect to formalize it in the near future. One of the intentions of this new consortium is not to be an exclusive club." He invited interested organizations to enter into discussions as to how they might join this new venture.

Further reflecting the federal government's confidence in high-performance computing, Wong said the administration was requesting $803 million for the fiscal year beginning next October, an increase of 26 percent over the figure for the current year.

Winners simulated 'dark matter.' "Most of the mass of the universe is in a form that we cannot see and about which we know very little," Salmon and Warren explained. "Astrophysicists call this material 'dark matter.' Exactly how large a fraction of the universe's mass is dark matter is the subject of intense study. We know it is there because we can observe its gravitational influence on the 'normal matter' that we *can* see."

Salmon and Warren are studying the evolution of dark matter in the universe. Its elusiveness is, in small measure, a blessing when it comes to studying its behavior. By definition, dark matter can influence itself, or other kinds of matter, only through gravity. Thus, simulation need only account for the gravitational interactions of bodies of dark matter. These interactions are called "N-body" simulations.

Since the summer of 1987, following his junior year at Caltech, Warren has been programming the algorithm that runs this simulation to operate on parallel processors. At the end of that summer he managed to get it to run on four processors. By the summer of 1988 it was running on 256 processors at Caltech. About a year later he and Salmon conducted their first experiment: two galaxies interacting with each other. Unfortunately, shortly into the simulation, even before the galaxies interacted, one of them ex-

Gary Smaby projects 35 percent market growth

The massively parallel supercomputer market amounted to $270 million last year, said Gary Smaby, president, Smaby Group, analysts of the supercomputer marketplace. He expects exceptional growth — something like 35 percent compounded each year over the next five years. Machines like the Delta will be increasingly used in everyday applications.

The shift in emphasis from defense and espionage to more peaceful scientific pursuits was summed up succinctly on one of the slides Smaby used in his presentation: "From nukes and spooks to genes and greens." "Clearly we are going to see a switch in the source of dollars that drive this industry," he explained. "In the past 10 years something on the order of 70 percent of the dollars that drove this industry came from the US government, directly or indirectly. What we need now are new applications that will drive commercial users of these machines."

ploded. After a hard week of analysis, they found the error in the algorithm, not in the code.

Year by year the two scientists have been improving the program and running it on larger machines. In 1990 Warren plotted the power of the four successive machines they had used and found that they lay on an exponential line representing a factor-of-eight improvement every year. The following year they gained another factor of eight on a massively parallel Ncube system at Sandia National Laboratory. This year they achieved another factor of eight on the Delta.

Warren explained, "This rate of increase represents, since I started in 1987, a factor of 50,000 in computation. I now wonder what the next point is going to be. I wouldn't be surprised if in 1993 I could add another factor-of-eight point."

The need for an ever greater number of processors is demonstrated by the very large numbers involved in this N-body simulation. For instance, each unit of dark matter has a mass of 5 million suns, Salmon said. One visualization he showed was "derived from research simulating the evolution of 8.7 million gravitating bodies to study the formation and structure of galaxies in an expanding universe." Each unit of space in the computation is about 30 million light years across. "The entire image represents a region of about 7 megaparsecs, or about 2^{20} kilometers on a side."

"By running large N-body simulations, we have been able to learn about the shapes and dynamics (motions) of the agglomerations of dark matter that form as a result of mutual gravitational interactions," the two astrophysicists said. "These agglomerations are called 'halos,' and they are believed to surround galaxies like the Milky Way. Dark halos are the incubators in which galaxies — and all the more familiar celestial objects like clusters, stars, planets, and nebulas — form. Understanding the structure and dynamics of dark halos is a step toward understanding the origins of all the other more accessible structures in the universe."

European Community votes to adopt *sui generis* database law

Richard H. Stern

The European Community (EC) Commission, on January 29, 1992, adopted a proposal for a directive protecting the content of databases. The EC's Council of Ministers and the European Parliament have until September 1992 to modify the proposed directive, after which it is anticipated that the directive will become binding law within the EC. The proposed database law will be similar to US law in some ways but will also differ from US law in several significant ways.

Under the EC proposal, as under US law, the selection and arrangement of preexisting information in the database are protected by copyright insofar as they are "original" (not copied and not trivial or dictated by the nature of the material). In general, however, computer databases have no *arrangement*, although there may be a selection of data fields. The selection of data fields for a database may or may not be trivial or functionally dictated. The selection of information to include in a database is sometimes 100 percent of what exists. In that case, US law does not protect the "selection"; at times, the selection may be too obvious to receive protection under US law. It is unclear how EC law will treat such cases.

The EC proposal diverges widely from current US law in creating a right against "unfair extraction." Under US law, the facts in a database are unprotected. Anyone is free to use the facts to prepare another work, even one competing with the work from which the facts are taken, as long as the second party does not take any of the first party's original elements of selection or arrangement. US law does not protect the "sweat equity," or expenditure of "sweat of the brow," that a person who compiled a database invested in doing so.

In contrast, article 2(5) of the EC proposal creates a 10-year legal right against unfair extraction or commercial reuse of material in a database. The facts compiled in a database will enjoy this new form of protection, even when they are not "original." In explaining this move, the EC said that this protection is necessary to protect database compilers from "misappropriation of the results of the . . . investment incurred in obtaining and collecting data, even when [unoriginal]." This protection is subject, however, to compulsory licensing where the material in the database cannot be created, collected, or obtained from any other source.

Thus, after 1993, electrical engineers in the EC will presumably have to be able to prove that they know by memory the values of π and e to however many decimal places they need, or else face liability for unfair extraction. Perhaps the same principle applies to ASCII codes and identification of interrupt vectors.

Finally, the EC proposal contains a reciprocity provision aimed at the US and patterned on the US Semiconductor Chip Protection Act of 1984. Article 11 makes unfair extraction rights available within the EC to foreign creators of databases only if their own countries offer similar protection to EC-based creators of databases. In other words, a US database seller is protected against unfair extraction in EC countries only if the US enacts an unfair extraction law.

The question in the US will then become whether the gain to US database sellers who operate in Europe will justify imposing protection of the facts in databases on all US users of facts in databases. Should US engineers either memorize the data in the *Handbook of Physics and Chemistry* or start paying to use the data in order to protect the *Handbook*'s publisher from unfair extraction in Europe? How far the principle would be extended is unclear. Assembly code mnemonics? Pin-outs for chips? Could a first chip-manufacturer compiler of a pin-out and parameter database for the chips that it introduced to the market prevent would-be second sources from unfairly extracting the data to make a form-, fit-, and function-compatible chip? Would the compulsory-licensing provision assure against this?

The EC proposal is evidence that despite the *Paperback* and *Accolade* decisions, and despite the "look and feel" cases, the US lower courts do not have a monopoly on wacky ideas about creating new intellectual property rights to plague electrical engineers.

Richard H. Stern writes the Micro Law column in *IEEE Micro*. His address is Oblon, Spivak, McClelland, Maier & Neustadt, P.C., 1755 Jefferson Davis Highway, Suite 400, Arlington, VA 22202.

STANDARDS

Editor: Norman F. Schneidewind, Naval Postgraduate School, Code AS/Ss, Monterey, CA 93943, fax (408) 646-3407, e-mail 0422p@cc.nps.navy.mil

Software requirements specifications standard undergoing revisions

The first in a possible series of meetings to complete revisions of the eight-year-old *IEEE Standard for Software Requirements* (ANSI/IEEE 830) for balloting will be held this summer, according to the committee chair, Edward R. (Ted) Byrne.

The meeting is planned for July or August at a computer-related conference convenient for the attendees, Byrne said. The draft will be distributed for ballot as soon as possible after the meeting or meetings.

Byrne said the information in the existing standard is now 10 years old and predates many major areas of interest — for example, the popularity of personal computers.

A standard cannot just grow old gracefully, he said. Instead, it must be reaffirmed, revised, or withdrawn. Hence, the IEEE has authorized and encouraged review of the standard.

For the past year, those with a close interest in the document have submitted their ideas to Byrne, and several draft revisions have been created and distributed. The general belief is that the standard should be revised to both strengthen it and include several new topics. Byrne said the time has arrived to widen the exposure and to prepare for a ballot on a revised Standard 830.

Byrne said the proposed draft differs from the 1984 standard in a number of ways in that it

- has been promoted from a guide to a recommended practice;
- addresses new aspects of requirements, such as object-oriented analysis and window-based applications;
- discusses the relationship between requirements and prototyping;
- gives several formats of requirements that are appropriate under different circumstances;
- removes tutorial information that is now outdated or more generally available;
- references other new standards that have appeared in the past eight years; and
- has undergone editorial improvement.

Still further revisions are possible, Byrne said.

He added that anyone who would like to participate in the revision process may request a copy of the draft for review and/or attend the meeting or meetings that will be held to finalize the draft document for balloting.

IEEE members who want to join the ballot group, but not necessarily be involved before the balloting, may submit their names for inclusion on a distribution list.

Contact Edward R. Byrne at Flatland Computer Specialities Inc., PO Box 430, Madison, NJ 07940-0430, phone (201) 822-3219.

Standards reference guide published

The World of Standards, which publisher 88open Consortium calls the industry's first standards reference guide, has been released. The guide contains more than 75 standards obtained from 19 participating international organizations, including the IEEE, the American National Standards Institute, the International Consultative Committee for Telephone and Telegraph (CCITT), and X/Open.

The guide describes each standard and lists the organization credited with its development. In addition, contact information, including name, address, and phone and fax numbers, is provided.

Copies may be ordered by dialing (408) 436-6642.

Guide for developing software life cycle processes planned

IEEE Working Group P1074.1 (Guide for Developing Software Life Cycle Processes) has been authorized to produce a companion document to IEEE-Std-1074-1991, *Standard for Developing Software Life Cycle Processes*.

The purpose of the guide is to assist anyone who intends to develop or maintain software in accordance with the standard. Once finalized, the guide will provide insight into the standard, elaborate on its key concepts, and address some of the major challenges users of the standard face.

The working group is preparing a draft of the guide that will be distributed to interested parties for review and comment. Dennis Nickle, working group chair, says he expects the guide to go to ballot in early 1994.

The working group's next two meetings are scheduled for October 12-14, 1992, in Portland, Oregon, and February 8-10, 1993, in Orlando, Florida.

Anyone who'd like to attend the meetings, or comment on the existing material, should contact Dennis Nickle, Q500 FC, E-Systems, Melpar Division, 7700 Arlington Blvd., Falls Church, VA 22046, phone (703) 849-1559.

To receive announcements and minutes of future meetings, contact David Schultz, Computer Sciences Corp., 15245 Shady Grove Rd., Rockville, MD 20850-3222, phone (301) 921-3090.

CALL FOR PAPERS

Call for articles and referees for *Computer*

Computer seeks articles for inclusion in future special issues.

Heterogeneous processing will be the theme of the **June 1993** issue. Manuscripts surveying or reporting original research, design and development, and applications are sought immediately in

- Network orchestration tools
- Profiling and matching
- Virtual heterogeneous machine experiences
- Heterogeneous processing programming and debugging environments
- Use and design of mixed-mode parallel machines
- Performance and cost-effectiveness strategies
- Heterogeneous processing as an educational and training paradigm

A 300-word abstract of each manuscript is due by **August 15, 1992**; 14 copies of each full manuscript must be submitted by **September 6, 1992**; notification of decisions is set for **January 4, 1993**; and the final version of each manuscript is due **March 1, 1993**.

For this special issue, direct submittals and questions to Richard F. Freund, NRaD, Code 423, 271 Catalina Blvd., San Diego, CA 92152-5000, phone (619) 553-4071, fax (619) 553-3926, e-mail freund@superc.nosc.mil. Questions may also be directed to H.J. Siegel, School of Electrical Engineering, 1285 Electrical Engineering Bldg., Purdue University, West Lafayette, IN 47907-1285, phone (317) 494-3444, fax (317) 494-6440, e-mail hj@ecn.purdue.edu.

Visualization will be the theme of the **July 1994** issue. Manuscripts surveying or reporting original research, design and development, and applications are sought immediately in

- Visualization methods
- Volume visualization
- Survey or tutorial on visualization applications
- Visualization resources and strategies
- User interfaces and interactive techniques for visualization
- Applications of visualization
- Hardware and workstations for visualization
- Software for visualization

A 300-word abstract of each manuscript is due by **June 15, 1993**; 14 copies of each full manuscript must be submitted by **August 1, 1993**; notification of decisions is set for **January 15, 1994**; and the final version of each manuscript is due **March 1, 1994**.

Submittals and questions for this special issue should be directed to Arie E. Kaufman, Dept. of Computer Science, State University of New York at Stony Brook, Stony Brook, NY 11794-4400, phone (516) 632-8441, e-mail ari@cs.sunysb.edu.

For submittal to Computer, *manuscripts must not have been previously published or currently submitted for publication elsewhere. Each manuscript should be no more than 32 typewritten, double-spaced, single-sided pages long, including all text, figures, and references. Each of the 14 copies submitted should include a page that contains the title of the article, the full name(s) and affiliation(s) of the author(s), complete postal and electronic address(es) of all the authors as well as their telephone and fax number(s), a 300-word abstract, and a list of keywords identifying the central issues of the manuscript's contents. The final manuscript should be approximately 8,000 words in length and contain no more than 12 references.*

If you are willing to review articles for any special issue, please send a note listing your research interests to Jon T. Butler, editor-in-chief of Computer, *at the Dept. of Electrical and Computer Engineering, Naval Postgraduate School, Code EC/Bu, Monterey, CA 93943-5004, Internet butler@ece.nps.navy.mil.*

IEEE Trans. on VLSI Systems will begin quarterly publication in January 1993, under the cosponsorship of the IEEE's Computer Soc., Circuits and Systems Soc., and Solid-State Circuits Council. Coverage will focus on VLSI/ULSI systems integration. Submit seven copies of the complete manuscript (up to 25 double-spaced pages and up to three pages of figures) to Sung-Mo (Steve) Kang, Editor-in-Chief, Coordinated Science Lab, Univ. of Illinois, 1101 W. Springfield Ave., Urbana, IL 61801-3082, phone (217) 244-0577, fax (217) 244-1653, e-mail kang@uivlsi.csl.uiuc.edu.

Usenix Winter 1993 Technical Conf.: Jan. 25-29, 1993, San Diego, Calif. Submit extended abstract by **July 20, 1992**, and final paper by **Nov. 20, 1992**, to Usenix Conf. Office, 22672 Lambert St., Suite 613, El Toro, CA 92630, phone (714) 588-8649, fax (714) 588-9706, e-mail conference@usenix.org.

ICSEE 93, Int'l Conf. on Simulation in Eng. Education: Jan. 17-20, 1993, La Jolla, Calif. Cosponsors: Soc. for Computer Simulation et al. Submit four copies of regular paper (about 12 double-spaced pages) or short paper (about 6 double-spaced pages) by **July 31, 1992**, and camera-ready copy by **Sept. 30, 1992**, to Charles Knadler, IBM, Advanced Automation Systems, 9201 Corporate Blvd., Rockville, MD 20850, phone (301) 640-3124, fax (301) 640-2136, e-mail knadlerc@wmavm7.vnet.ibm.com.

12th World Congress: July 18-23, 1993, Sydney, Australia. Sponsor: Int'l Federation of Automatic Control. Submit five copies of paper by **July 31, 1992**, and final paper by **Feb. 28, 1993**, to IFAC Secretariat, Electrical and Computer Eng. Dept., Univ. of Newcastle, University Dr., Callaghan, NSW 2308, Australia, phone 61 (49) 601-712, fax 61 (49) 216-025, e-mail eegcg@cc.newcastle.edu.au.

ICBGM 93, 1993 Int'l Conf. on Bond Graph Modeling and Simulation: Jan. 17-20, 1993, San Diego, Calif. Sponsor: Soc. for Computer Simulation. Submit short paper (about six pages) or long paper (about 12 pages) by **July 31, 1992**, and camera-ready version by **Sept. 30, 1992**, to Jose J. Granda, Mechanical Eng. Dept., California State Univ. at Sacramento, Sacramento, CA 95819, phone (916) 278-5711, fax (916) 278-5949, e-mail grandajj@ecs.csus.edu; or Francois E. Cellier, Electrical and Computer Eng. Dept., Univ. of Arizona, Tucson, AZ 85721, phone (602) 621-2692, fax (602) 621-8076, e-mail cellier@ece.arizona.edu.

1993 Conf. on Discrete and Combined Simulation: Jan. 17-20, 1993, La Jolla, Calif. Sponsor: Soc. for Computer Simula-

tion. Submit four copies of regular paper (about 12 double-spaced pages) or short paper (about 6 double-spaced pages) by **July 31, 1992**, and camera-ready copy by **Sept. 30, 1992,** to Stanislaw Raczynski, Univ. Panamericana, Esoula de Ingenieria, Augusto Rodin 498, 03910 Mexico D.F.; or SCS, 4838 Ronson Ct., Suite L, San Diego, CA 92111, phone (619) 277-3888, fax (619) 277-3930, e-mail mcleod@scsc.bitnet.

1993 Conf. on Simulation in the Health Sciences and Services: Jan. 17-20, 1993, La Jolla, Calif. Sponsor: Soc. for Computer Simulation. Submit four copies of regular paper (about 12 double-spaced pages) or short paper by **July 31, 1992**, and camera-ready copy by **Sept. 30, 1992,** to Meyer Katzper, 2 Locks Pond Ct., Rockville, MD 20854, phone (301) 443-0316, fax (301) 443-9283, e-mail $m1k@pccjes2.bitnet.

Fifth Workshop on Software Reuse: Oct. 26-29, 1992, Palo Alto, Calif. Cosponsor: IEEE Computer Soc. Technical Committee on Software Eng. Submit position paper in required format by **Aug. 1, 1992**. For submission guidelines, contact Martin Griss, Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304-1126, phone (415) 857-8715, fax (415) 857-8526, e-mail griss@hpl.hp.com.

ISUMA 93, Second Int'l Symp. on Uncertainty Modeling and Analysis: Apr. 25-28, 1993, College Park, Md. Cosponsors: Univ. of Maryland et al. Submit four copies of 250-word abstract by **Aug. 1, 1992**, and final paper by **Jan. 15, 1993**, to Bilal M. Ayyub, Civil Eng. Dept., Univ. of Maryland, College Park, MD 20742, phone (301) 405-1956, fax (301) 314-9320, e-mail ba15@umail.umd.edu.

1992 Int'l Bankai Workshop on Adaptive Intelligent Systems: Oct. 12-14, 1992, Brussels. Sponsor: Soc. for Worldwide Interbank Financial Telecommunications. Submit abstract by **Aug. 1, 1992**, to Brigitte Diraison, AI Lab, SWIFT s.c., 1 Av. Adèle, 1310 La Hulpe, Belgium, phone 32 (2) 655-3153, fax 32 (2) 655-3785.

Working Conf. on Architectures and Compilation Techniques for Fine- and Medium-Grain Parallelism: Jan. 20-22, 1993, Orlando, Fla. Sponsor: Int'l Federation for Information Processing. Submit four copies of 20-page manuscript by **Aug. 14, 1992**, and camera-ready paper by **Oct. 30, 1992**, to Jean-Luc Gaudiot, Univ. of Southern California, Electrical Eng.-Systems Dept., Los Angeles, CA 90089-2563, phone (213) 740-4484, fax (213) 740-4449, e-mail gaudiot@usc.edu.

IEEE Workshop on Imprecise and Approximate Computation: Dec. 1, 1992, Phoenix, Ariz. Cosponsors: IEEE Computer Soc. Technical Committee on Real-Time Systems et al. Submit six copies of position paper (up to five pages) by **Aug. 15, 1992**, and camera-ready version by **Nov. 6, 1992**, to Wei Zhao, Computer Science Dept., Texas A&M Univ., College Station, TX 77843-3112, phone (409) 845-5098, fax (409) 847-8578, e-mail zhao@cs.tamu.edu.

Third Int'l Workshop on Network and Operating System Support for Digital Audio and Video: Nov. 12-13, 1992, La Jolla, Calif. Cosponsor: IEEE Comm. Soc. Submit 500- to 2,000-word position paper or extended abstract by **Aug. 15, 1992**, by e-mail to av-workshop@cs.ucsd.edu. Submit final paper by **Oct. 15, 1992**, to P. Venkat Rangan, Multimedia Lab, Computer Science Dept., Univ. of California at San Diego, La Jolla, CA 92093-0114, phone (619) 534-5419, fax (619) 534-7029, e-mail av-workshop@cs. ucsd.edu or venkat@cs.ucsd.edu.

J. of Computer and Software Eng. seeks papers on various topics and applications for its issues No. 1 and 2 of 1994. Publisher: Ablex. Submit five copies of full manuscript by **Aug. 21, 1992,** to E.K. Park, Computer Science Dept., US Naval Academy, Annapolis, MD 21402, phone (410) 267-3037, fax (410) 267-4883, e-mail eun@usna.navy.mil.

OE/Technology 92, Optical Tools for Advanced Manufacturing and **Optical Technologies and Applications:** Nov. 25-20, 1992, Boston. Sponsor: Int'l Soc. for Optical Eng. Submit abstract by **Aug. 24, 1992**, and manuscript by **Oct. 19, 1992** to SPIE, PO Box 10, Bellingham, WA 98227-0010, phone (206) 676-3290, fax (206) 647-1445.

CAIA 93, Ninth IEEE Conf. on Artificial Intelligence for Applications: Mar. 1-5, 1993, Orlando, Fla. Submit four copies of short paper (up to 1,000 words) with extended abstract or long paper (up to 5,000 words) with 200-word abstract by **Aug. 31, 1992**, and final paper by **Dec. 14, 1992**, to David Waltz, Thinking Machines Corp., 245 1st St., Cambridge, MA 02142-1214, phone (617) 234-2050, fax (617) 234-4444, e-mail waltz@think.com.

Solid Modeling 93, Second ACM/IEEE Symp. on Solid Modeling and Applications: May 19-21, 1993, Montreal, Canada. Submit abstract (150-300 words) by **Sept. 1, 1992**, full paper (6,000-word maximum) by **Oct. 15, 1992,** and camera-ready version by **Feb. 7, 1993**, to Mary Johnson, Design Research Center, CII 7015, Rensselaer Polytechnic Inst., Troy, NY 12180-3590, phone (518) 276-6751, fax (518) 276-2702, e-mail mjohnson@rdrc.rpi.edu.

Int'l J. of Computer Simulation seeks manuscripts for a special issue on simulation metamodeling of production systems. Publisher: Ablex. Submit five copies of full paper by **Sept. 1, 1992**, to Christian N. Madu, Management Science Dept., Lubin

Call for articles and referees for IEEE Computer Society magazines

The following IEEE magazines seek articles for inclusion in future issues:

IEEE Design & Test of Computers seeks general-interest articles (20 double-spaced pages, including figures) for its 1993 issues. Suggested topics include reviews of tools, design experiences, and design methodologies; use of CAE/CAD tools in the design process; test strategies and economics; DFT vs. cost; D/A test; system-level trade-offs; rapid prototyping; packaging; manufacturing databases; and special-purpose tools. Submit a 300-word abstract, keywords, and your mail/e-mail addresses, along with six copies of the article, to Manuel d'Abreu, GE R&D, 1 River Rd., PO Box 8, KW-C308, Schenectady, NY 12301, phone (518) 387-7097, fax (518) 387-7332, dabreu@crd.ge.com.

IEEE Micro invites authors to submit original articles for the April 1993 joint issue with *Computer* on multichip modules (*Micro* will address packaging and interconnections) by **Aug. 1, 1992**. Also welcome at any time are general-interest submissions. Suggested topics include multiprocessing, optical and biological computing, microcomputing to aid the handicapped, systems design, DSP-based tools, solid-state memory, and fuzzy-logic chips. Submit six copies of each article to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C. so duca degli Abruzzi, 24, 10129 Torino, Italy, e-mail delcorso@polito.it. For author guidelines, dial (714) 821-8380.

IEEE Software seeks articles for inclusion in a special issue on software synthesis to appear in May 1993. Reusability using software synthesis techniques, domain-specific software synthesis, and handling of cost trade-offs in software synthesis are examples of the topics of interest. Eight copies of previously unpublished articles should be sent by **Sept. 1, 1992,** and final versions are due by **Jan. 15, 1993.** Make submittals to Dorothy Setliff, 341 Benedum Hall, Electrical Eng. Dept., Univ. of Pittsburgh, Pittsburgh, PA 15261, phone (412) 624-9695, fax (412) 624-1108, e-mail setliff@ee.pitt.edu.

Graduate School of Business, Pace Univ., 1 Pace Plaza, New York, NY 10038, phone (212) 346-1919 or 1980, fax (212) 346-1872.

SIGCSE 93, 24th Technical Symp. on Computer Science Education: Feb. 18-19, 1993, Indianapolis, Ind. Sponsor: ACM. For paper submittal requirements, contact Cary Laxer or Frank Young, Rose-Hulman Inst. of Tech., Terre Haute, IN 47803, phone (812) 877-8401, e-mail laxer@cs.rose-hulman.edu. Submittal deadline: **Sept. 1, 1992**.

EDAC 93, Fourth European Design Automation Conf., and **EuroASIC 93, European Event in ASIC Design:** Feb. 22-25, 1993, Paris. Submit paper by **Sept. 2, 1992**, and final version of manuscript by **Nov. 27, 1992,** to EDAC-EuroASIC 93, CEP Consultants, 26-28 Albany St., Edinburgh, EH1 3QH, Scotland, phone 44 (31) 557-2478, fax 44 (31) 557-5749.

ETC 93, European Test Conf.: Apr. 19-24, 1993, Rotterdam, The Netherlands. Submit four copies of paper by **Sept. 11, 1992**, and camera-ready manuscript by **Jan. 22, 1993,** to Jacques Kevers, ETC 93 Secretariat, IEEE Computer Soc., 13 Avenue de l'Aquilon, B-1200 Brussels, Belgium, phone 32 (2) 770-22-42 or 21-98, fax 32 (2) 770-85-05, e-mail j.kevers@compmail.com.

Third Int'l Symp. on Database Systems for Advanced Applications: Apr. 6-8, 1993, Deajon, Korea. Sponsors: Korean Information Science Soc., Information Processing Soc. of Japan. Submit six copies of full paper (5,000-word maximum) by **Sept. 12, 1992**, to SongChun Moon, 207-43 Cheongryang, Dongdaemun, Seoul 130-012, Korea, fax 82 (2) 960-6743, e-mail moon@dbsun2.kaist.ac.kr.

26th Simulation Symp.: Mar. 29-Apr. 1, 1993, Washington, D.C. Sponsor: Soc. for Computer Simulation. Submit four copies of complete paper (10 to 20 double-spaced pages) by **Sept. 15, 1992**, to John A. Miller, Computer Science Dept., 415 Graduate Studies Research Center, Univ. of Georgia, Athens, GA 30602, phone (706) or (404) 542-3440, e-mail jam@pollux.cs.uga.edu.

IWSR 93, Second Int'l Workshop on Software Reusability: Mar. 24-26, 1993, Lucca, Pisa, Italy. Cosponsors: ACM SIGSoft et al. Submit five copies of full paper by **Sept. 15, 1992**, and camera-ready version by **Dec. 15, 1992**, to Bill Frakes, Software Eng. Guild, 400 Drew Ct., Sterling, VA 22170, phone (703) 450-5954, e-mail 70761.1176@CompuServe.com.

CompEuro 93: May 24-27, 1993, Paris-Evry, France. Submit paper by **Sept. 15, 1992**, and final manuscript by **Feb. 15, 1993**, to CompEuro 93, c/o Société des Electriciens et des Electroniciens, 48 rue de la Procession, F-75724 Paris, Cedex 15, France, phone 33 (1) 44-49-60-60, fax 33 (1) 44-49-60-44.

CALENDAR

In the accompanying Calendar and adjoining Call for Papers, the IEEE Computer Society logo identifies the conferences, symposiums, and workshops the society is sponsoring or cooperating in. Other multiple-topic events of interest to our readers are also listed. The notices are published free of charge, in chronological order, and as space permits.

For inclusion in the Call for Papers section, please submit the name of the event, date(s), location, sponsor(s), deadline for submittals, phone and fax numbers, and — if applicable — the electronic address of the person to whom papers should be directed.

For the Calendar section, please provide the event name, date(s), location, sponsor(s), phone and fax numbers, and the electronic address of the person to contact for complete information.

Submittals should arrive at *Computer* **at least five weeks before the month of publication** (i.e., for the **September 1992** issue, send information for receipt by **July 20, 1992**) to Chuck Governale, Calendar Dept., *Computer*, PO Box 3014, Los Alamitos, CA 90720-1264, fax (714) 821-4010, e-mail c.governale@compmail.com.

July 1992

ICS 92, Sixth ACM Int'l Conf. on Supercomputing, July 19-24, Washington, D.C. Sponsor: ACM SIGArch. Contact Ken Kennedy, CITI Rice Univ., PO Box 1892, Houston, TX 77251, phone (713) 527-6009, e-mail ken@rice.edu; Constantine Polychronopoulos, CSRD, Univ. of Illinois, 104 S. Wright St., Urbana, IL 61801, phone (217) 244-4144, e-mail cdp@csrd.uiuc.edu; Harry Wijshoff, Computer Science Dept., Utrecht Univ., Padualaan 14, 3584 CH Utrecht, The Netherlands, e-mail harryw@cs.ruu.nl; or Toshitugu Yuba, Electrotechnical Lab, 1-1-4 Umezono, Tsukuba, Ibaraki 305, Japan, e-mail yuba@etl.go.jp.

1992 Int'l Symp. on Optical Applied Science and Eng., July 19-24, San Diego, Calif. Sponsor: Int'l Soc. for Optical Eng. Contact SPIE, PO Box 10, Bellingham, WA 98227-0010, phone (206) 676-3290, fax (206) 647-1445.

Logic at Tver 92: Logical Foundation of Computer Science, July 20-24, Tver, Russia. Cosponsors: IEEE et al. Contact M.A. Taitslyn, Univ. of Tver, 33 Zhelyahova St., Tver 170013, Russia.

1992 Complex Systems Eng. Synthesis and Assessment Tech. Workshop, July 20-24, Silver Spring, Md. Sponsor: Naval Surface Warfare Center. Contact Steve Howell, NSWC, Code U33, 10901 New Hampshire Ave., Silver Spring, MD 20903-5000, phone (301) 394-3987, fax (301) 394-1175, e-mail showell@nswc-wo.navy.mil.

First Joint European Conf. on Information Systems, July 22-24, London. Sponsors: ACM et al. Contact Elias Awad, Univ. of Virginia McIntire School of Commerce, Charlottesville, VA 22903, phone (804) 924-3423.

Coling 92, 14th Int'l Conf. on Computational Linguistics, July 23-28, Nantes, France. Sponsor: Int'l Committee on Computational Linguistics. Contact A. Zampolli, Univ. di Pisa, ILC, via della Faggiola 32, I-56100, Pisa, Italy, phone 39 (50) 560-481, fax 39 (50) 589-055.

Siggraph 92, 19th Int'l Conf. on Computer Graphics and Interactive Techniques, July 26-31, Chicago. Sponsor: ACM. Contact Siggraph 92, 401 N. Michigan Ave., Chicago, IL 60611, phone (312) 321-6830, fax (312) 321-6876, e-mail info92@siggraph.org.

Colt 92, Fifth ACM Workshop on Computational Learning Theory, July 27-29, Pittsburgh. Contact Robert Daley or Betty Brannick, Computer Science Dept., 322 Alumni Hall, Univ.of Pittsburgh, Pittsburgh, PA 15260, phone (412) 624-5930 or 8493, e-mail daley@cs.pitt.edu or brannick@cs.pitt.edu.

ISSAC 92, Int'l Symp. on Symbolic Algebraic Computation, July 27-29, Berkeley, Calif. Sponsor: ACM. Contact Erich Kaltofen, Rensselaer Polytechnic Inst., Computer Science Dept., Troy, NY 12180-3590, phone (518) 276-6907, e-mail kaltofen@cs.rpi.edu.

Broadband Analog and Digital Opto-Electronics, July 29-31, Santa Barbara, Calif. Sponsor: IEEE Lasers and Electro-Optics Soc. Contact IEEE LEOS, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331, phone (908) 562-3893, fax (908) 562-1571.

August 1992

17th Congress Workshop, Aug. 2-14, Washington, D.C. Sponsor: United Nations. Contact Lawrence W. Fritz, GE Aerospace, PO Box 8048-10A26, Philadelphia, PA 19101, phone (215) 531-3205, fax (215) 962-3698.

Conf. on Artificial Intelligence and Symbolic Math Computations, Aug. 3-6, Karlsruhe, Germany. Contact Jacques Calmet, Univ. of Karlsruhe, Fakultät für Informatik, Am Fasanengarten 5, Postfach 6980, D-7500 Karlsruhe 1, Germany, phone 49 (0721) 608-4208, fax 49 (0721) 69-68-93.

ECAI 92, 10th European Conf. on Artificial Intelligence, Aug. 3-7, Vienna. Sponsor: Austrian Soc. for Artificial Intelligence. Contact ECAI 92 Conf. Office, ADV, Trattnerhof 2, A-1010 Vienna, Austria, phone 43 (1) 533-0913-74, fax 43 (1) 533-0913-77.

Space Operations, Applications, and Research Symp., Aug. 4-6. Houston. Cosponsors: US Air Force Material Command, NASA Johnson Space Center. Contact Kumar Krishen, NASA JSC, Houston, TX 77058, phone (713) 283-5875.

ASAP 92, Int'l Conf. on Application-Specific Array Processors, Aug. 4-7, Berkeley, Calif. Sponsor: Univ. of California at Berkeley. Contact Jose Fortes, School of Electrical Eng., Purdue Univ., West Lafayette, IN 47907, phone (317) 494-3646, fax (317) 494-6440, e-mail fortes@ecn.purdue.edu; or Mary Stewart, EECS Dept., Cory Hall, Univ. of California, Berkeley, CA 94720.

Workshop on Image Processing with Khoros, Aug. 4-7, Albuquerque, N.M. Sponsor: Univ. of New Mexico College of Eng. Contact College of Eng., Univ. of New Mexico, Professional Eng. Development, Workshop Programs, Farris Eng. Center, Rm. 131, Albuquerque, NM 87131-1387, phone (505) 277-6061 or (800) 292-7051, fax (505) 277-0813.

Hot Chips IV, Symp. on High-Performance Chips, Aug. 9-11, Stanford, Calif. Sponsor: IEEE Computer Soc. Technical Committee on Microprocessors and Microcomputers. Contact Glen Langdon, Univ. of California at Santa Cruz, Santa Cruz, CA 95064, phone (408) 459-2212, fax (408) 429-0146, e-mail langdon@cse.ucsc.edu.

35th Midwest Symp. on Circuits and Systems, Aug. 9-12, Washington, D.C. Cosponsors: George Washington Univ. et al. Contact Mona E. Zaghloul, Electrical Eng. and Computer Science Dept., George Washington Univ., Washington, DC 20052, phone (202) 994-3772, fax (202) 994-0227, e-mail zaghloul@seas.gwu.edu.

High-Speed Optoelectronic Devices and Circuits, Aug. 9-14, Banff, Alta., Canada. Sponsor: Eng. Foundation. Contact Eng. Foundation, 345 E. 47th. St., New York, NY 10017, phone (212) 705-7835, fax (212) 705-7441.

PODC 92, 11th Symp. on Principles of Distributed Computing, Aug. 10-12, Vancouver, B.C., Canada. Sponsor: ACM. Contact Norm Hutchinson, Univ. of British Columbia, Computer Science Dept., 6356 Agriculture Rd., Vancouver, B.C. V6T 1Z2, phone (604) 822-8188, e-mail hutchinson@cs.ubc.ca.

Fifth Usenix C++ Conf., Aug. 10-13, Portland, Ore. Contact Usenix Conf. Office, 22672 Lambert St., Suite 613, El Toro, CA 92630, phone (714) 588-8649, fax (714) 588-9706, e-mail conference@usenix.org.

4CCCG, 4th Canadian Conf. on Computational Geometry, Aug. 10-14, St. John's, Nfld., Canada. Contact Cao An Wang, Computer Science Dept., Memorial Univ. of Newfoundland, St. John's, Nfld., Canada A1C 5S7, phone (709) 737-4590, fax (709) 737-2009, e-mail 4cccg@cs.mun.edu.

LUV 92, Lisp Users and Vendors Conf., Aug. 10-14, San Diego, Calif. Sponsors: Assoc. of Lisp Users et al. Contact Laura Lotz, An Event to Remember, PO Box 294, Malvern, PA 19355-0294, phone (215) 651-2990, fax (215) 651-0936.

Third Systems Reengineering Tech. Workshop, Aug. 11-13. Contact Tamra Moore, Code U33, Naval Surface Warfare Center, 10901 New Hampshire Ave., Silver Spring, MD 20903-5000, phone (301) 394-1913, fax (301) 394-1175, e-mail tmoorc@nswc-wo.navy.mil.

Crypto 92, Aug. 16-20, Santa Barbara, Calif. Sponsor: Int'l Assoc. of Cryptologic Research. Contact Spyros S. Magliveras, Computer Science and Eng. Dept., Univ. of Nebraska, Lincoln, NE 68588-0115, phone (402) 472-5005, fax (402) 472-7767, e-mail spyros@helios.unl.edu.

SIGComm 92, Aug. 17-20, Baltimore. Sponsor: ACM. Contact J.J. Garcia-Luna, SRI Int'l, 333 Ravenswood Ave., Menlo Park, CA 94025, fax (415) 859-6028, e-mail sigcomm92-info@ihburn.att.com.

ICPP 92, 21st Int'l Conf. on Parallel Processing, Aug. 17-21, St. Charles, Ill. Sponsor: Pennsylvania State Univ. Contact Tse-yun Feng, Penn State Univ., ECE Dept., EE East Bldg., University Park, PA 16802, phone (814) 863-1469, fax (814) 865-7065, e-mail tfeng@ecl.psu.edu.

Computer-Based Design Environments Symp.-Workshop, Aug. 18-19, Baden-Baden, Germany. Contact Jens Pohl, CAD Research Unit, Cal Poly, San Luis Obispo, CA 93407, fax (805) 756-5986.

Int'l Workshop on Distributed Object Management, Aug. 19-21, Edmonton, Alta., Canada. Cosponsors: Canadian Information Processing Soc. et al. Contact M. Tamer Ozsu, Computing Science Dept., Univ. of Alberta, Edmonton, Alta., Canada T6G 2H1, phone (403) 492-2860, fax (403) 492-1071.

Int'l Conf. on Intelligent Systems Eng., Aug. 19-21, Edinburgh, UK. Sponsor: Institution of Electrical Engineers. Contact IEE Conf. Services, Savoy Place, London WC2R 0BL, UK, phone (071) 240-1871, fax (071) 497-3633.

VLDB 92, 18th Int'l Conf. on Very Large Data Bases, Aug. 23-27, Vancouver, B.C., Canada. Cosponsors: VLDB Endowment, Canadian Information Processing Soc. et al. Contact Paul Sorenson, Computer Science Dept., Univ. of Alberta, Edmonton, Alta., Canada T6G 2H1, phone (403) 492-4589, fax (403) 492-1071, e-mail vldb92@cs.ualberta.ca or sorenson@cs.ualberta.ca.

September 1992

Workshop on Neural Networks — Techniques and Applications, Sept. 7-8, Liverpool, England. Cosponsors: IEEE United Kingdom-Republic of Ireland Computer Chapter et al. Contact Katrina Houghton, Computer Science Dept., Univ. of Liverpool, Liverpool L69 3BX, England, UK, phone 44 (51) 794-3668, fax 44 (51) 794-3715.

Euro DAC 92, European Design Automation Conf., Sept. 7-10, Hamburg, Germany. Cosponsors: IEEE Circuit and Systems Soc. et al. Contact MP Associates, 7490 Clubhouse Rd., Suite 102, Boulder, CO 80301, phone (303) 530-4562, fax (303) 530-4334; or A. Zylbersztejn, Bull SA, 121 Ave. De Malakoff, 75116 Paris, France, phone 33 (14) 502-9583, fax 33 (14) 502-9307.

ICIP 92, Second Singapore Int'l Conf. on Image Processing, Sept. 7-11, Singapore. Cosponsors: IEEE Singapore Section et al. Contact Technical Programme Chair, ICIP 92, IEEE Singapore Section, 200 Jalan Sultan, #11-03 Textile Centre, Singapore 0719, phone (65) 291-9690, fax (65) 292-8596.

HPDC I, First Int'l Symp. on High-Performance Distributed Computing, Sept. 9-10, Syracuse, N.Y. Cosponsor: Syracuse Univ. Contact Geoffrey C. Fox, NPAC, Syracuse Univ., Rm. 3-131, 111 College Pl., Syracuse, NY 13244-4100, phone (315) 443-4741, fax (315) 443-1973, e-mail hpdc@ nova.npac.syr.edu.

UPPL 92, Univ. Parallel Processing Lab Workshop, Sept. 10-11, Alta, Utah. Contact Lyle Bingham, Computer System Architects, 100 Library Plaza, 15 N. 100 E., Provo, UT 84606-3100, phone (801) 374-2300, fax (801) 374-2306.

LCN 92, 17th Conf. on Local Computer Networks, Sept. 13-16, Minneapolis. Sponsor: IEEE Computer Soc.

Technical Committee on Computer Comm. Contact Marc Cohn, Raychem, MS 123/6612, 300 Constitution Dr., Menlo Park, CA 94025-1164, phone (415) 361-3902, fax (415) 361-6248; or Steve Bell, Hughes LAN Systems, MS 392, 1072 S. Saratoga Svale Rd., San Jose, CA 95129, phone (415) 966-7926, fax (415) 966-7990, e-mail sbell@hls.com.

Fifth Digital Signal Processing Workshop, Sept. 13-16, Starved Rock State Park, Ill. Sponsor: IEEE Signal Processing Soc. Contact Mark J.T. Smith, School of Electrical Eng., Georgia Inst. of Tech., Atlanta, GA 30332-0250, phone (404) 894-6291.

SESS, Software Eng. Standards Symp., Sept. 13-17, Brighton, UK. Contact Takis Katsoulakos, Lloyd's Register, Lloyd's Register House, 29 Wellesley Rd., Croydon, CR0 2AJ, UK, phone (081) 681-4774.

DCCA 92, Third Working Conf. on Dependable Computing for Critical Applications, Sept. 14-16, Mondello, Italy. Sponsor: Int'l Federation for Information Processing Working Group 10.4. Contact Carl E. Landwehr, Code 5540, Naval Research Lab, Washington, DC 20375-5000, phone (202) 767-3381, fax (202) 404-7942; or Luca Simoncini, Dipartimento di Ingegneria dell'Informazione, Univ. of Pisa, Via Diotisalvi 2, 56100 Pisa, Italy, phone 39 (50) 593-443 or 550-100, fax 39 (50) 554-342, e-mail simon@icnucevm.cnuce.cnr.it.

ICARCV 92, Second Int'l Conf. on Automation, Robotics, and Computer Vision, Sept. 15-18, Singapore. Cosponsors: Singapore Institution of Engineers et al. Contact ICARCV Secretariat, Associated Conventions and Exhibitions, 204 Bukit Timah Rd., #04-00, Boon Liew Bldg., Singapore 0922, phone (65) 799-5470, fax (65) 791-2687, e-mail emital@ntuvax.bitnet.

IEEE Workshop on Visual Languages, Sept. 15-18, Seattle. Contact Tadao Ichikawa, Faculty of Eng., Hiroshima Univ., 1-4-1 Kagamiyama, Higashi-Hiroshima 724, Japan, phone 81 (824) 227-029, fax 81 (824) 227-195.

WLI, IEEE Conf. on Wireless LAN Implementation, Sept. 17-18, Dayton, Ohio. Contact James T. Geier, 685 N. Enon Rd., Yellow Springs, OH 45387, phone (513) 255-6224.

KBSE 92, Seventh Knowledge-Based Software Eng. Conf., Sept. 20-23, Tysons Corner, Va. Sponsor: Rome Lab. Contact W. Lewis Johnson, USC Information Sciences Inst., 4676 Admiralty Way, Marina del Rey, CA 90292-6695, phone (310) 822-1511, fax (310) 823-6714, e-mail johnson@isi.edu; or Barbara Radzisz, Data and Analysis Center for Software, PO Box 120, Utica, NY 13503, phone (315) 336-0937, kbse7-request@cs.rpi.edu.

ITC 92, 1992 Int'l Test Conf., Sept. 20-24, Baltimore. Cosponsor: IEEE Philadelphia Section. Contact Doris Thomas, 514 Pleasant Valley Blvd., Suite 3, Altoona, PA 16602, phone (814) 941-4666, fax (814) 941-4668.

ASIC 92, Fifth IEEE Int'l Conf. on Application-Specific Integrated Circuits, Sept. 21-25, Rochester, N.Y. Sponsor: IEEE Rochester Section. Contact Lynne M. Engelbrecht, 170 Mt. Read Blvd., Rochester, NY 14611, phone (716) 328-2310, fax (716) 436-9370; or Y. Tim Tsia, Eastman Kodak, MC-02015, 1669 Lake Ave., Bldg. 81.454/KRL, Rochester, NY 14650-2015, phone (716) 722-4896, fax (716) 722-9053.

Compsac 92, 16th Int'l Computer Software and Applications Conf., Sept. 22-25, Chicago. Contact Stephen Yau, Univ. of Florida, CIS Dept., Rm. 301, Gainesville, FL 32611, phone (904) 335-8006.

PRICAI, Second Pacific Rim Int'l Conf. on Artificial Intelligence, Sept. 23-25, Sungdong-ku, Korea. Cosponsors: Korea Information Science Soc. et al. Contact Jim Hyung Kim, Korea Advanced Inst. for Science and Tech., Center for Artificial Intelligence Research, 373-1 Kusong-dong, Yusong-ku, Taejon, 305-701, Korea, phone 82 (42) 829-3517, fax 82 (42) 829-8700, e-mail jkim@cair.kaist.ac.kr.

DT 92, Int'l Conf. on Data Transmission, Sept. 23-25, London. Sponsor: Institution of Electrical Engineers. Contact Alan Clark, Dowty Communications, Mayze House, Westmead, Swindown SN5 7YT, phone 44 (0793) 511-789, fax 44 (0793) 511-683.

IWOOOS 92, Int'l Workshop on Object Orientation in Operating Systems, Sept. 24-25, Paris. Sponsor: Institut National de Recherche en Informatique et en Automatique. Contact Roy Campbell, Univ. of Illinois, Computer Science Dept., 2413 Digital Lab, 1304 W. Springfield, Ave., Urbana, IL 61801, phone (217) 333-0215 or 3328, fax (217) 333-3501, e-mail roy@cs.uiuc.edu or eric.diku.dk.

13th Int'l Symp. on Electronics Manufacturing Tech., Sept. 28-30, Baltimore. Cosponsors: IEEE Components, Hybrids, and Manufacturing Tech. Soc., Electronic Industries Assoc. Contact Bill Moody, 2529 Eaton Rd., Wilmington, DE 19810, phone (302) 478-4143, fax (302) 478-7057.

Graphicon, Computer Graphics in Science and Arts, Sept. 28-Oct. 3, Moscow, Russia. Cosponsor: ACM. Contact Yury Golubev, Keldysh Inst. of Applied Math, Miusskaya Sq. 4, Moscow 125047, Russia, phone (095) 250-7817.

Int'l Workshop on Hardware-Software Codesign, Sept. 30-Oct. 2, Estes Park, Colo. Cosponsors: IEEE Computer Soc. et al. Contact Joanne E. DeGroat, Ohio State Univ., 205 Dreese Lab, 205 Neil Ave., Columbus, OH 43210, phone (614) 292-2439, e-mail degroat@ee.eng.ohio-state.edu; or Alfred E. Dunlop, AT&T Bell Labs, 600 Mountain Ave., Murray Hill, NJ 07974, phone (908) 582-6570, e-mail aed@allegra.att.com.

October 1992

Second Int'l Workshop on Responsive Computer Systems, Oct. 1-2, Saitama, Japan. Sponsor: US Office of Naval Research. Contact Miroslaw Malek, Electrical and Computer Eng. Dept., Univ. of Texas, Austin, TX 78712, phone (512) 471-5704, fax (512) 471-0954, e-mail malek@cmx.utexas.edu; or Tohru Kikuno, Information and Computer Science Dept., Osaka Univ., 1-1, Machikaneyama-cho, Toyonaka-shi, Osaka, 560, Japan, phone 81 (6) 844-1151, fax 81 (6) 841-9741, e-mail kikuno@ics.osaka-u.ac.jp.

Sixth Software Eng. Inst. Conf. on Software Eng. Education, Oct. 5-6, and **11th SEI Educator Development Workshop, Oct. 7,** San Diego, Calif. Cosponsor: Software Eng. Inst. Contact Carol Sledge, SEI, Carnegie Mellon Univ., Rm. 4206, Pittsburgh, PA 15213-3890, phone (412) 268-7708, fax (412) 268-5758, e-mail cas@sei.cmu.edu.

11th Symp. on Reliable Distributed Systems, Oct. 5-7, Houston. Cosponsors: IEEE Computer Soc. Technical Committee on Distributed Processing et al. Contact Thomas F. Lawrence, Rome Lab/C3AB, Bldg. 3, Griffiss Air Force Base, NY 13441, phone (315) 330-2158, fax (315) 330-3911; or Kishor S. Trivedi, Electrical Eng. Dept., Duke Univ., Durham, NC 27706, phone (919) 660-5269, e-mail kst@egr.duke.edu.

ISSRE 92, Third Int'l Symp. on Software Reliability Eng., Oct. 7-9, Research Triangle Park, N.C. Cosponsors: IEEE Computer Soc. Technical Committee on Software Eng. et al. Contact Mladen A. Vouk, Computer Science Dept., Box 8206, North Carolina State Univ., Raleigh, NC 27695-8206, phone (919) 515-7886, fax (919) 515-5839, e-mail vouk@adm.csc.ncsu.edu.

Milcom 92, Oct. 11-14, San Diego, Calif. Cosponsors: IEEE Comm. Soc. et al. Contact Milcom 92, General Dynamics Electronics Div., PO Box 85106, San Diego, CA 92186-5106; Security Dept. MZ 7101-G, Milcom 92, General Dynamics Electronics Div., PO Box 85227, San Diego, CA 92186-5227; or Dennis E. Litchfield, Office of the Assistant Secretary of Defense (C^3I), The Pentagon, Rm. 3D 228, Washington, DC 30301-3040.

ICCD 92, Int'l Conf. on Computer Design, Oct. 12-14, Cambridge, Mass. Cosponsor: IEEE Circuit and Systems Soc. Contact IEEE Computer Soc., 1730 Massachusetts Ave. NW, Washington, DC 20036-1903, phone (202) 371-1013, fax (202) 728-0884.

ASPLOS 5, Fifth Conf. on Architectural Support for Programming Languages and Operating Systems, Oct. 12-15, Boston. Sponsor: ACM. Contact Barry Flahive, Hewlett-Packard, Apollo Systems Div., 300 Apollo Dr., MS CHR 02 DE, Chelmsford, MA 01824, phone (508) 256-2471 or 6600, e-mail flahive@apollo.hp.com.

Seventh Int'l Software Process Workshop, Oct. 15-18, Yountville, Calif. Sponsor: Rocky Mountain Inst. of Software Eng. Contact Ian Thomas, Software Design and Analysis, 444 Castro St., Suite 413, Mountain View, CA 94041, phone (415) 694-1464.

Frontiers 92, Fourth Symp. on the Frontiers of Massively Parallel Computation, Oct. 19-21, McLean, Va. Cosponsors: NASA Goddard Space Flight Center et al. Contact Pearl Wang, Computer Science Dept., George Mason Univ., 440 University Dr., Fairfax, VA 22030-4444, phone (703) 993-1527, fax (703) 993-1521.

Visualization 92, Oct. 19-23, Boston. Sponsor: IEEE Computer Soc. Technical Committee on Computer Graphics. Contact Bruce E. Brown, Oracle, 500 Oracle Pkwy., Box 659412, Redwood Shores, CA 94065, phone (415) 506-6202, fax (415) 726-0983; or Gregory M. Nielson, Arizona State Univ., Rural Rd. and University Ave., Tempe, AZ 85287, phone (602) 965-2785, e-mail nielson@enuxva.eas.asu.edu.

FOCS, Foundations of Computer Science, Oct. 25-27, Pittsburgh. Contact Gary Miller, School of Computer Science, Carnegie Mellon Univ., Pittsburgh, PA 15213-3890, phone (412) 268-2631 or (412) 268-2526.

26th Asilomar Conf. on Signals, Systems, and Computers, Oct. 26-28, Pacific Grove, Calif. Cosponsors: Naval Postgraduate School et al. Contact Neil K. Jablon, AT&T Bell Labs, 200 Laurel Ave., Rm. 1G-540, Middletown, NJ 07748-4801, phone (908) 957-2011, fax (908) 957-7943, e-mail nkj@mtqub.att.com.

Fifth Workshop on Software Reuse, Oct. 26-29, Palo Alto, Calif. Cosponsor: IEEE Computer Soc. Technical Committee on Software Eng. Contact Martin Griss, Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304-1126, phone (415) 857-8715, fax (415) 857-8526, e-mail griss@hpl.hp.com.

Fourth NASA Symp. on VLSI Design, Oct. 29-30, Coeur d'Alene, Idaho. Cosponsors: Spokane, Washington, IEEE Section, Univ. of Idaho NASA Space Eng. Research Center. Contact Sterling Whitaker, NASA SERC, Univ. of Idaho, Moscow, ID 83843, phone (208) 885-6500, fax (208) 885-6840, e-mail swhitaker@grieg.mrc.uidaho.edu.

November 1992

ISCIS 7, Seventh Int'l Symp. on Computer and Information Sciences, Nov. 2-4, Antalya, Turkey. Cosponsors: IEEE Computer Soc. Turkey Chapter et al. Contact Erol Gelenbe, Ecole des Hautes Etudes en Informatique, 45 rue des Saints-Peres, 75006 Paris, France, phone 33 (1) 4286-2136, fax 33 (1) 4286-2231, e-mail erol@ehei.ehei.fr; or Nese Yalabik, Computer Eng. Dept., Middle East Technical Univ., 06531 Ankara, Turkey, phone 90 (4) 223-7100, ext. 2079, fax 90 (4) 286-8624.

DFT 92, 1992 IEEE Int'l Workshop on Defect and Fault Tolerance in VLSI Systems, Nov. 4-6, Dallas. Contact D.M.H. Walker, Electrical and Computer Eng. Dept., Carnegie Mellon Univ., Pittsburgh, PA 15213-3890, phone (412) 268-8522, fax (412) 268-2860, e-mail dmw@ece.cmu.edu.

Workshop on High-Level Synthesis, Nov. 4-6, Dana Point, Calif. Contact Wolfgang Rosensteil, Forschumrtzentrum Informatik, Germany, phone 49 (721) 6906-401, fax 49 (721) 6906-409.

3rd IFIP Working Conference on

DEPENDABLE COMPUTING FOR CRITICAL APPLICATIONS

Can we rely on computers?

September 14-16, 1992 - Mondello, Sicily, Italy

This is the third Working Conference on this topic, following the successful conferences held in August, 1989, at Santa Barbara (USA) and in February, 1991, at Tucson (USA). As evidenced by papers that were presented and discussed at those meetings, critical applications of computing systems are concerned with differing service properties, relating to both the nature of proper service and the system's ability to deliver it. These include thresholds of performance and real-time responsiveness that demark loss of proper service (failure), continuity of proper service, ability to avoid catastrophic failures, and prevention of deliberate privacy intrusions.

As a Working Conference, the program has been designed in order to promote the exchange of ideas by extensive discussions. The sessions are entitled: Functional Testing, Specification and Verification of Fault-Tolerance, Dependability and Performance, Application of Formal Methods, Online Error Detection, Safety Critical Industrial Systems, Experimantal Evaluation, and Protocols for Dependability. All the paper sessions will end with a 30 minute discussion period on the topics dealt with in the session. In addition to the paper sessions, three panel sessions have been organized. The first, entitled "Safe Vehicle-Highway Systems" will explore safety requirements, design methods and validation techniques for computing and communication subsystems associated with intelligent vehicle-highway systems. The second, entitled "Malicious and Inadvertent Human Operator Faults" will explore current and proposed techniques for detecting and countering faults introduced by the human operator. The third, entitled "Security Issues in Intelligent Networks" will deal with privacy problems related to the delivery of intelligent network services and related customer control, along with network security problems mostly related to open network provisioning.

Advance registration as well as early hotel reservation is required. No on-site registration will be available.
The Provisional Program as well as the Registration and Hotel Reservation Forms in e-mail format are available by the General Chairman Luca Simoncini, e-mail address: SIMON@IET.UNIPI.IT or SIMON@ICNUCEVM.CNUCE.CNR.IT

CAREER OPPORTUNITIES

RATES: $15.00 per line (ten lines minimum). Average five typeset words per line, eight lines per column inch. Add $10 for box number. Send copy at least one month prior to publication date to: **Marian B. Tibayan, Classified Advertising, COMPUTER Magazine, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, CA 90720-1264; (714) 821-8380; fax: (714) 821-4010.**

In order to conform to the Age Discrimination in Employment Act and to discourage age discrimination, *COMPUTER* may reject any advertisement containing any of these phrases or similar ones: "...recent college grads...," "...1-4 years maximum experience...," "...up to 5 years experience," or "...10 years maximum experience." *COMPUTER* reserves the right to append to any advertisement without specific notice to the advertiser. Experience ranges are suggested minimum requirements, not maximums. *COMPUTER* assumes that since advertisers have been notified of this policy in advance, they agree that any experience requirements, whether stated as ranges or otherwise, will be construed by the reader as minimum requirements only. *COMPUTER* encourages employers to offer salaries that are competitive, but occasionally a salary may be offered that is significantly below currently acceptable levels. In such cases the reader may wish to inquire of the employer whether extenuating circumstances apply.

D.E. SHAW & CO.
Algorithmic Trading

D.E. Shaw & Co. is a small (several dozen employees), highly capitalized (over $300 million in partners' equity), extremely successful Wall Street firm specializing in quantitative finance and computational trading. Computer scientists and system designers form the professional core of the firm, and not merely its support staff, and participate in its profits. Our technical staff includes both Ph.D.-level researchers recruited from Stanford, MIT, and other leading computer science departments and extraordinarily talented B.S.-and M.S.-level system designers and "superhackers". It is our practice to compensate unusually gifted individuals at a level exceeding that of the market. Applicants should send resumes to The Recruiting Department, D.E. Shaw & Co., 39th Floor, Tower 45, 120 W. 45th St., New York, NY 10036.

PROGRAMMER ANALYST

Will be responsible for all aspects of maintenance and development of large computer system for oilfield diving service company. Specific duties include daily backup of system, user support, file size maintenance, daily maintenance and upgrade, set up and maintenance of local area network; maintenance and developmental programming for employer. Developmental programming includes complex data base systems. Will assume full responsibility for all personal computer (PC) hardware and software; duties will include PC hardware maintenance and integrity, and customization of all software applications. M.S. in Computer Science required. Any academic background or experience with development of data base applications software; 12 credit hours in data base systems, data structures, operating systems, and/or data base management systems (any combination). Must be willing to work in rural area. Relocation expenses not paid by employer. Salary: $26,000/year, 7AM to 4PM, 40 hours/week. Contact LA Office of Employment Security, Job Order 456672, 706 E. Vermilion, Lafayette, LA 70502.

RESEARCH SYSTEMS ANALYST
(Project Manager)

40 hrs/wk, 8-5, $34,700/yr. Data and systems analysis and database design and administration for inventory management POS and information systems. Design and maintain detailed documentation for all departmental mainframe and PC applications. Perform adhoc financial and statistical data analysis and generate related reports for various areas of the company. Application programming on IBM 3090 mainframe under an MVS environment using JCL, ROSCOE, TSO and SAS. PC application programming using DOS, PC SAS and DBase III on IBM PC compatible. Use LOTUS III, Harvard Graphics and Microsoft Word to develop presentations and enhance analysis. Minimum requirements: MBA with a major field of study or concentration in Information Systems. 1 year in the design of a comprehensive management information system using structured analysis/design and database design techniques and 6 months in DBase III programming. 9 college credit hours in finance and 9 college credit hours in statistics. Apply at the Texas Employment Commission, Dallas, Texas, or send resume to the Texas Employment Commission, TEC Building, Austin, Texas 78778, J.O. #6521675. Ad Paid by an Equal Opportunity Employer.

SOFTWARE DESIGN ENGINEER

Design software from functional specifications; code, test, and debug the product; and develop documentation for software maintenance, user interface and product release. Minimum requirements: M.S. in Computer Science with concentration in Wordprocessing software; proficiency in C and Assembler languages and in Relational Data Base Management Systems (RDBMS). 40 hrs/wk. $34,000/yr. The job order number for this job opportunity is KS3305352. Please apply at Wichita Department of Human Resource Office, 402 E. 2nd, P.O. 877, Wichita, KS 67201-0877, Telephone Number (316) 266-8600 or refer to job order number when submitting a resume to the above referenced office. Do not submit resumes to Alien Certification Officer.

DEVELOPMENT ENGINEER

Development Engineer for testing and support development for Sequelink, writing test scripts/cases, executing tests, data analysis, problem analysis, reporting, tracking. Salary for a 40 work week 9a-5p, M-F is $32,000 yearly. Applicant with M.S. degree in Electrical Engineering and 1 year experience in job or Graduate Research Assistant with knowledge of PS/2, AS400, RS6000, VAX, SUN, MacINTOSH, OS/2, NOVELL Netware, DOS, WINDOWS, UNIX, VMS, OS/2 DBMS, ORACLE, INGRESS, INFORMIX, Rdb, Sybase, SQL Server send resumes only to: Job Service of Florida, 2660 W. Oakland Park Blvd., Ft. Lauderdale, FL 33311-1347, Attn: Job Order # FL 0612315.

UNITED STATES MILITARY ACADEMY
Visiting Professor in Computer Science

Applications are invited for AY 1992-1993 for appointment consideration under the provisions of the Intergovernmental Personnel Act of 1979. Applicants must have Ph.D., possess strong commitment to undergraduate teaching, and currently hold the academic appointment of professor or associate professor. Duties include teaching in, and providing advice on, the Academy's computer science programs. Compensation is usually commensurate with that currently being received. Send resumes to Colonel Daniel M. Litynski, Department of Electrical Engineering and Computer Science, United States Military Academy, West Point, NY 10996. Applications must be received by 1 October 1992. The US Military Academy is an affirmative action/equal opportunity employer.

THE UNIVERSITY OF MELBOURNE
AUSTRALIA
Chair of Computer Engineering

The University of Melbourne invites applications for a Chair of Computer Engineering which has been established in the Department of Electrical and Electronic Engineering. The Department together with the Department of Computer Science, forms the School of Information Technology and Electrical Engineering.

The Department has extensive teaching and research programs in electrical and electronic engineering. Its research activities include software reliability, computer architecture and networking, distributed processing, photonics, communications, signal processing, control and biomedical engineering.

The successful applicant will have a distinguished record of professional achievement and will be committed to developing and active research and teaching program. It is anticipated that the successful applicant will form strong research links with industry.

A base salary is $A73,800 per annum on appointment and will rise to $A77,900 from July 1992.

Further information about the position, application procedures, conditions for outside work, superannuation, travel and removal expenses, housing assistance and conditions of appointment, is available from the Registrar. All correspondence (marked "Personal and Confidential") should be addressed to the Registrar, The University of Melbourne, Parkville, Victoria, 3052, Australia.

Telephone (613) 344 7529(Ms Kaye Goldenberg). Facsimile (613) 344 6897. Applications close on 30 September, 1992.

The Council reserves the right to make no appointment or to fill the Chair by invitation at any stage.

The University of Melbourne is an equal opportunity employer and has implemented a smoke-free work-place policy.

AUBURN UNIVERSITY
Earle C. Williams
Eminent Scholar Chair
In Electrical Engineering

Nominations and applications are invited for the Earle C. Williams Eminent Scholar Chair in Electrical Engineering. Candidates for this chair should have achieved national and international prominence in digital systems and/or microelectronics.

Applicants or nominees must have an earned doctorate, senior academic experience, and a documented record of distinction in university teaching and research. The successful candidate will be expected to provide intellectual leadership in his/her area of expertise for the Department of Electrical Engineering as well as enrich the scholarly environment at Auburn University.

Auburn University is located in the city of Auburn in east-central Alabama. This land-grant university enrolls more than 21,000 students, the largest on-campus enrollment in the state. The Department of Electrical Engineering, one of eight departments within the College of Engineering, offers Bachelor, Master, Master of Science and Ph.D. degrees in Electrical Engineering. The department has a current enrollment of 939 undergraduate students and 100 graduate students. The 28 full-time faculty have an annual research expenditure of approximately $2 million.

The Search Committee will begin its review of applications immediately. Interested candidates should submit: (1) a detailed resume, (2) a letter indicating an interest in the chair, the candidate's academic philosophy, and a brief statement of accomplishments in teaching and research, and (3) names and addresses of five references. Nominations should be submitted with the complete name, mailing address and telephone number of the individual nominated.

Applications and nominations should be sent to Professor J. David Irwin, Department of Electrical Engineering, Auburn University, AL 36849-5201. Auburn University is an affirmative action/equal opportunity employer. Applications from minority and female candidates are encouraged.

UNITED STATES
AIR FORCE ACADEMY
Department of Computer Science
Visiting Faculty Position

The Department of Computer Science of the United States Air Force Academy invites applications for a Visiting Professor/Associate Professor/Assistant Professor position. We seek qualified applicants with extensive experience teaching computer science and a record of scholarly activity. Duties will include teaching undergraduate courses, reviewing our academic program, and promoting undergraduate and faculty research. Applicant must be a U.S. citizen and should have a demonstrated commitment to undergraduate computer science programs. Applicant must be currently affiliated with an institution of higher education or be a local, state, or federal employee. The appointment is normally for one year and will begin July 1993. Salary is commensurate with your current salary level. To apply, please send your vita by 1 September 1992 to: Chairman, Department of Computer Science, United States Air Force Academy, USAF Academy, CO 80840-5701.

SYSTEMS ANALYST

Will be responsible for planning, analyzing, designing, coding, testing, and implementing new computer software applications as well as major enhancements to existing systems/applications in electric utility environment. Will serve from time to time as project leader on systems/applications; perform post critiques on newly implemented systems; evaluate software/hardware or services as needed. Will also provide support for maintenance of production software applications. M.S. or B.S. Computer Science required with experience in database design, Microsoft Windows, PC Hardware, multi-computer networks, Unix, MS-DOS, and "C". Must be able to work with wide range of computer languages and have excellent academic record with courses taken in Data Communication and Networks, Computer Aided Design, Microcomputer Hardware and Applications, Parallel Computer Architecture, and Electrical Engineering. At least 3 credit hours or equivalent in each subject area above. $30,156-36,000/year. 8AM to 5PM; 40 hr/week. Contact Mississippi State Employment Office. P.O. Box 1720, 2229 22nd St., Gulfport, MS 39502-1720 JON MS2612082.

YONSEI UNIVERSITY
Department of Computer Science

The department invites applications for faculty positions in the area of system softwares, programming languages, compilers, algorithms, and theory of computation. Appointments begin March '93 but September '93 appointments are also possible. A candidate must have a Ph.D. in computer science or related fields and must have at least one year experience by the appointment date. Please call Professor J.S. Song at +82-2-361-2714 (Fax +82-2-365-2579) no later than the end of August '92 or send a resume to Professor J.S. Song, Department of Computer Science, Yonsei University, Seoul, 120-749, KOREA. Korean language proficiency is required.

ACCOUNTING SYSTEMS ANALYST

Accounting Systems Analyst is needed to work for a Grain Company in Edinburg, Texas. Responsibilities are: Analyze, develop, test, debug, install and maintain individual computerized financial systems. Conduct daily surveys of commodity prices and operations to establish current and correct cost information to implement fully integrated cost accounting system. Evaluates and installs new software packages, provides technical support to end users, implements strategies for remote processing of data to branch locations using multiplexers, and interfaces with various vendors in the development of an integrated in house system. Provide management with financial information such as intercompany and bank reconciliation statements, general ledger including documentation, monthly cash flow statements, and information for annual budget, physical inventory and audit statements. Assist company CPA with preparation of tax reports and audit reports. Use a 4th generation computer language to generate departmental management expense reports. Salary $27,807.00 per year, work Monday through Friday 8:00 A.M. to 5:00 P.M. Must have M.B.A., Major Field of Study Accounting. Must have 1 year experience as Computer Systems Analyst. Must submit resume. Apply at the Texas Employment Commission, Austin, Texas, J.O. #6687436. "Ad Paid by an Equal Employment Opportunity Employer."

CSC NASA
Research Opportunities
Parallel and Distributed Computing

Computer Sciences Corporation (CSC), a major NASA contractor, has opportunities in the areas of distributed computing and software tools for parallel computers. These openings are in the Numerical Aerodynamic Simulation (NAS) Systems Division of our Applied Research Branch, which operates two Cray computers, a 32K processor Connection Machine CM-2 and 128 processor Intel iPSC/860. Each researcher has dedicated access to a Silicon Graphics IRIS 4D/25 workstation. One important Applied Research Branch mission is to perform state-of-the-art research in parallel scientific computation for computational aerosciences. Areas of interest are:

1. Distributed computing (emphasizing heterogeneous systems) including distributed algorithms, load sharing and distributed sharing, data representation, data communication and network protocols, and performance analysis. Expertise in these areas should be applicable for solving computational aerospace applications across a network of high performance computers.

2. Software tools including programming languages—in particular, extensions to FORTRAN 77; automatic partitioning; scheduling and dynamic load balancing techniques; optimizing compilation techniques; program development tools; and performance monitoring tools. These tools should be designed for high performance on aerospace applications.

We are seeking candidates who have a Ph.D. in Computer Science, Computer Engineering, Electrical Engineering, or a related discipline; a strong research background; and interest in the above areas. Other candidates with extensive practical experience will also be considered. CSC offers a competitive salary, comprehensive benefits, travel opportunities, access to state-of-the-art computational facilities, and collaboration in very active and advanced scientific computing research programs. Send your resume, referencing number RNR150, to: Computer Sciences Corporation, Attn: Darla Cowden, Human Resources, P.O. Box 390757, Mountain View, CA 94039. All resumes must be received by September 1, 1992. An Equal Opportunity Employer M/F/D/V.

INDUSTRIAL ENGINEERING ANALYSTS

Two or more Industrial Engineering Analysts required. Design, creation, coding and implementation of simulation models using SIMAN/Cinema, C and FORTRAN in workstation and microcomputing environment (UNIX, DOS/OS2 and VMS/VAX). Perform extensive statistical data analysis using SAS. Develop and implement mathematical models and optimization algorithms for sequencing and scheduling decision support systems using OR techniques of Linear Programming, Dynamic Programming, Network Analysis and Queing Theory. Facilities planning including design, development and implementation of material handling systems for improved operational efficiency using knowledge of manufacturing/production systems. Prepare reports and Technical documentation for all work performed. Applicants required to have a Masters Degree or its equivalent in Industrial Engineering with at least one year experience in the job offered. Graduate Research/Project work will be acceptable to satisfy experience requirements. Salary will be $36,000.00/year for a 40 hour work week. Must have proof of legal authority to work in the U.S. Interested applicants apply at the Texas Employment Commission, Ft. Worth, Texas, or send resume to the Texas Employment Commission, Austin, Texas 78778-0001, Job Order No. 6521683. This advertisement was paid by an Equal Opportunity Employment employer.

NETWORK ENGINEER

40 hrs/wk, 9-6, $28,960/yr. Engineer the design and implementation of Local Area Network. Upgrading Network. Testing new Network Communication Technologies. Provide technical support for customers' LANS. Troubleshooting and problem identification. Provide on-site training in installation and verification of new system software releases, systems administration, security administration, and operations support. Testing/quality control of computer products. Evaluation of customers' systems requirements and integration of computer systems. Computer aided material and inventory management of hardware, software, and computer related equipment parts. Minimum requirements: Masters in Computer Science. 1 year in computer aided material management and quality control and in computer system hardware troubleshooting. 3 graduate credit hours in computer networks and 6 college credit hours in electronics or industrial electronics. Past administration of a Netware v2.2 network, diagnosing, troubleshooting, and upgrading NetWare LANS, and installation of both v2.2 and v3.11 operating systems. Apply at the Texas Employment Commission, Arlington, Texas, or send resume to the Texas Employment Commission, TEC Building, Austin, Texas 78778, J.O. #6687430. Ad Paid by an Equal Opportunity Employer.

SENIOR SYSTEMS ANALYST

Direct the activities of project team members to design and develop custom application programs (for internal use only) using the Progress Relational Database and Application Development System and C on a variety of minicomputers running assorted implementations of the Unix Operating System. Coordinate activities with department heads (clients) and participate in the product design, the preparation of technical specifications and design documents, programming and testing processes. Manage the activities of team members, assigning and prioritizing tasks, reviewing programs and design documents for compliance with technical specifications, and answering technical questions. Investigate new hardware and software products. Coordinate field tests. Make recommendations to senior management regarding purchases. Keep abreast of industry developments in the areas of hardware, relational databases, and graphical user interfaces. Requires a Master's Degree in Computer Science or its equivalent, and two years experience in job offered or two years directly related computer programming/analysis experience. Included within the two years experience, must have one year of UNIX system administration and maintenance functions experience. Requires one year experience working with the TCP/IP network protocol, the establishment of a multi-node, heterogeneous device network and performance tuning, and Sequent Symmetry multiprocessing computer in addition to installation/programming experience with the X-windowing system. 40 hour work week. $36,100.00 per year. Apply at the Texas Employment Commission, Dallas, Texas or send resume to the Texas Employment Commission, TEC Building, Austin, Texas 78778. Job Order #6687372. Ad paid by an Equal Employment Opportunity Employer.

THE WICHITA STATE UNIVERSITY
Chairperson
Computer Science Department

Applications are invited for the position of chairperson of the Computer Science Department. The closing date for applications is August 17, 1992 or the first of the month thereafter until the position is filled. Anticipated starting date is January 1, 1993 or July 1, 1993.

Required qualifications for the position include a Ph.D. in Computer Science or a closely-related discipline, an established record of Computer Science research and teaching at a Ph.D. granting department, and proven leadership. Salary will be commensurate with qualifications and experience.

The Computer Science Department has approximately 500 undergraduate majors and 120 graduate students. B.S., B.A., M.S., and M.C.S. degrees are currently offered with plans underway for offering the Ph.D. degree in Computer Science. A primary responsibility of the new chairperson will be the development of the Ph.D. program.

The department has 17 full-time faculty whose research interests include the areas of artificial intelligence, machine learning and discovery, automated theorem-proving, theoretical computer science, databases, logic programming, software engineering, programming languages, and simulation and modeling.

Letters of application including a resume and names and addresses of at least five references should be sent to: Dr. Rajshekhar Sunderraman, Chair of the Chair Search Committee, Box 83, The Wichita State University, Wichita, Kansas 67208. TEL: (316) 689-3156. FAX: (316) 689-3770. E-Mail: raj@cs.twsu.edu. The Wichita State University is an Equal Opportunity/Affirmative Action Employer.

SOFTWARE ENGINEER

Redesign and implement a generalized parser/translator in order to improve performance and stability. Support an associated grammar writing utility. Port grammar compilation and validation functions from Lisp to C. Assist in the design and implementation of other portions of current software product. Must have MS in Computer Science and 3 years experience as Software Engineer. Experience must include 1 year experience in C programming and Lisp programming; writing natural language parsers; porting software to a variety of hardware and software platforms; extensive knowledge of Unix. Programming experience on mainframe, minicomputer, microcomputer, and scientific workstations. Knowledge of Ross Improved Left-Corner Parsing algorithm. Must have academic specialization in computational linguistics. Must have at least 2 courses in computational linguistics and 2 courses in computational complexity and compiler theory. 45 hours per week. 9:00 a.m. to 6:00 p.m. Wage Range: $43,000 to $45,000 per year. Apply at the Texas Employment Commission, Austin, Texas, or send resume to the Texas Employment Commission, TEC Building, Austin, Texas 78778, J.O. #6687593. Ad Paid by an Equal Employment Opportunity Employer.

AIR FORCE INSTITUTE OF TECHNOLOGY
Computer Engineering/ Computer Science

The Electrical and Computer Engineering Department, School of Engineering, at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, Ohio. Applications are invited for a tenure track preferably at the assistant or associate professor level, effective immediately. We are seeking an individual with expertise in the artificial intelligence, software engineering, computer architecture, data base, or computer graphics specialty. Applicant must be a U.S. citizen and have an earned doctorate in computer or electrical engineering, computer science, or in a related specialty. Position requires teaching at the graduate level and research under the sponsorship of government agencies. This department has numerous close working relationships with Air Force and Department of Defense research and development organizations. In particular, the selected individual is expected to work closely with research and development organizations specializing in parallel computation and architectures, software engineering, and computer simulation. This department has excellent laboratory facilities in all of these areas. Computational facilities are of the highest caliber and are continually being expanded and advanced. Salary will be commensurate with experience. Submit a complete resume and the names of three references to Dr. John D'Azzo, Head, Dept. of Electrical and Computer Engineering, AFIT/ ENG, Wright-Patterson AFB, OH 45433-6583. The United States Air Force is an equal opportunity, affirmative action employer.

UNIVERSITY OF TENNESSEE
Position available at the Oak Ridge National Laboratory

The Computer Science Department at the University of Tennessee and the Mathematical Sciences Section at Oak Ridge National Laboratory have two postdoctoral research positions available in the areas of networking, computational science, and database/ filesystem management. The University of Tennessee and Oak Ridge National Laboratory are engaged in a DARPA-funded research project to develop a next-generation software distribution system. The new system is intended to be a successor to the NETLIB and XNETLIB software distribution systems. The principal investigators of the project are Jack Dongarra of the University of Tennessee and Oak Ridge National Laboratory, and Tom Rowan of Oak Ridge National Laboratory.

Familiarity with network computing, databases, X-window applications, parallel architectures, and scientific computing experience are desired. Benefits of the position include a competitive salary, travel opportunities, access to state-of-the-art computational facilities (including both parallel architectures and high-performance workstations), and collaborative research opportunities in a very active research program in advanced scientific computing.

Inquiries should be directed to: Jack Dongarra, Computer Science Department, University of Tennessee, Knoxville, TN 37996-1301; (615) 974-8295 (phone); (615) 974-8296 (fax); dongarra@cs.utk.edu (e-mail).

The University of Tennessee and Oak Ridge National Laboratory are equal opportunity/affirmative action employer.

COMPUTER SYSTEMS MANAGER II

Direct, coordinate and exercise functional authority over the activities of a technical support team involved in systems development, programming, software installation and Local Area Network (LAN) installations and operations. Conduct strategic planning to define and develop goals and objectives. Perform analysis of systems, hardware and software to determine long-range and short-term courses of action to meet defined objectives. Review the analysis of staff and assimilate data for senior management. Manage, design, oversee and implement end-user training, and technical and end-user documentation. Oversee the LAN environment including installation, testing and maintenance of hardware, software, dial-in system and security system; develop applications, policies and procedures; and coordinate all data transfers between systems. Supervise the installation of the CD-Rom file server and player to be used for Compustat and other databases. Requires a Bachelor's Degree in Computer Science or Business Computer Information Systems and five years experience in job offered or five years of directly related computer programming experience. Or will consider candidates with Master's Degree in Computer Science or Information Systems (or course work completed towards such degree), with the additional education substituting for up to one year of the required experience. Or will consider a Bachelor's Degree in Computer Science or Business Computer Information Systems and thirty (30) credit hours towards Master's Degree in Computer Science or Information Systems and fifty (50) months of directly related Computer Programming experience. Background must include two years of experience in the following: 1) LAN Management; 2) Design and implementation of LAN(s); 3) Computer Aided Software Engineering (CASE) tools and application software; 4) LAN Novell software (i.e. Fox PRO, C and C++, Windows, Nexpert, Commander); 5) Computer-aided software engineering and Expert Systems, 40 hour work week, $41,568.00 per year. Apply at the Texas Employment Commission, Denton, Texas or send resume to the Texas Employment Commission, TEC Building, Austin, Texas 78778. Job Order #6687647. Ad paid by an Equal Employment Opportunity Employer.

SOFTWARE ENGINEERING GROUP LEADER

The Jackson Laboratory and the Mouse Genome Informatics Project seek a Software Engineering Group Leader to oversee design and implementation of a database of genetic and physical mapping information, analytical routines, graphical interfaces, and display tools that support the use of the laboratory mouse as a model for human genetic diseases. This Project, which is a fully funded component of the NIH Human Genome Program, will establish a comprehensive, centralized information resource for the genetics research community. The scientist will be the senior computing scientist in a group including software engineers, geneticists, and molecular biologists. Applicants with a Ph.D. degree in computer science are strongly preferred. Broad technical experience in data management, relational and object-oriented databases, graphical user interfaces, and data visualization is required. Preference will be given to those who have managed software teams in a scientific environment. Application deadline: open until filled. Apply to Dr. Joseph H. Nadeau, Project Leader, Mouse Genome Informatics Project, Jackson Laboratory, 600 Main St., Bar Harbor, Maine 04609.

The Jackson Laboratory is an Equal Opportunity/Affirmative Action Employer.

SOFTWARE JOBS ABROAD

The International Computer Professional Associaton provides you with the worldwide contacts and up-to-the-moment information you need to find an exciting software assignment in Paris, London, and many other cities around the world. For more information about this new network of international opportunities call us at (415) 695-7618, 24 hours/day.

BOOK REVIEWS

Editor: Alan Kaminsky, Rochester Institute of Technology, PO Box 9887, Rochester, NY 14623-0887
(716) 475-5255; Internet ark@cs.rit.edu;Bitnet arkics@ritvax.

Hermes: A Language for Distributed Computing

Robert E. Strom, David F. Bacon, Arthur P. Goldberg, Andy Lowry, Daniel M. Yellin, and Shaula Alexander Yemini (Prentice Hall, Englewood Cliffs, N.J., 1991, ISBN 0-13-389537-8, 288 pp., $29)

Distributed computing is clearly a good thing, and the hope of getting more performance, higher reliability, and lower cost attracts more application designers to it every day. The question is how to program such a system to hide the ugliness of distribution (for example, transport protocols and the lack of shared memory) from users.

The Hermes language offers an approach that may be practical — and is certainly thought-provoking. Hermes has no pointers or shared memory, and byte orders are hidden from the programmer. Communication is message-based: Each input port connects to one or more output ports in a way that is data-type secure and supports user authentication. Access rights are supported on a port basis. For example, one output port may have only query rights at an input port whereas another output port has update rights.

A Hermes program is composed of many processes. A process is a sequential program that communicates only via messages. Hermes takes the purist position of using only messages for interprocess communication, partly because shared memory is expensive for general distributed systems. Additionally, the authors argue that Hermes programs tend to be more robust than those based on shared memory because the language enforces a discipline over interprocess interfaces, whereas shared memory allows one process to arbitrarily modify the state of another.

All distributed programming operations are fully integrated into the language: creating processes, connecting communications channels, and sending messages. The goal is to hide the system details from an application programmer, while making messages easy to use. I sympathize with readers who wonder how this goal can support high performance. In fact, it will be difficult to make a Hermes application as fast as an optimized C implementation. However, since communication is integrated into the language rather than added on as a remote-procedure-call package, the resulting code will likely be easier to maintain. In the long term, a language like Hermes may win over the speed of a pointer-based language like C, just as C has triumphed over assembly language.

The designers are conscious of the performance issue. At the process-management level, the Hermes compiler transforms a process context switch into a procedure call. At the data-structure level, the compiler provides efficient representations of high-level set constructs. The programmer may help the compiler with "pragmas" that do not change the meaning of a program but suggest data structures (in a way analogous to index selection in relational database systems). Hermes also uses copy-on-write set representations to reduce the cost of set-assignment operations.

Current research may lead to sophisticated integrated transformation tools. For example, a user may write a sequential program that performs a collection of transfers from one bank account to another. Hermes will transform this program into a parallel program that will abort a given transfer if funds in the source account are insufficient.

Hermes is an attractive language, and the book is very well written. The authors (all at the IBM T.J. Watson Research Center) are eager to interest users and will provide software to most researchers and professors. It's worth a look.

Dennis Shasha
Courant Institute, NYU
New York City

Data Networks, Second Edition

Dimitri Bertsekas and Robert Gallager (Prentice Hall, Englewood Cliffs, N.J., 1992, ISBN 0-13-200916-1, 556 pp., $52)

In the preface to this book, the authors write, "Our approach is . . . to provide a balance between the description of existing networks and the development of analytical tools." They succeed perfectly in maintaining this sensitive balance. Their book is a must-read for people interested in data networks. It offers thorough material on point-to-point protocols, delay models, multiaccess communications, routing, and flow control.

The reader is assumed to have some background in probability theory and some general knowledge in electrical engineering or computer science, but essentially the book is self-contained. Major concepts and principles are explained first in an informal, easy-to-understand way, followed by an in-depth description of modeling issues and mathematical analysis. Each chapter is supplemented with an extensive problem set, ranging from simple exercises to advanced problems that demand insight. The book is an ideal textbook for advanced graduate courses in which the general Open Systems Interconnection hierarchy is not the terminal but the starting point of study (as it is in the book). Thus, it wonderfully complements Andrew Tanenbaum's second edition of *Computer Networks* (Prentice Hall, 1988), which can be used at the undergraduate level.

Finally, let me cite the authors on the best use of the book:

> Although the analytical material can be used to analyze the performance of various networks, we believe that its more important use is in sharpening one's conceptual and intuitive understanding of the field; that is, analysis should precede design rather than follow it.

What else can I add? Reading this book should precede the beginning of any serious work in data networks rather than follow it.

Imrich Chlamtac
University of Massachusetts
Amherst, Massachusetts

Mirror Worlds

David Gelernter (Oxford University Press, New York, NY, 1991, ISBN 0-19-506812-2, 237 pp., $24.95)

What's past is prolog, and the prolog to information technology is so stunning that the future can be envisioned boldly. *Mirror Worlds* puts forth such a vision. It targets both the general and professional public. Its style is informal, making it easy, fast reading, even though Gelernter takes more pages than necessary to present his message.

A "mirror world" models a chunk of reality based on information technology. In a mirror world, real-time streams — no! oceans — of data and multimedia information are flowing, and you can select whatever is relevant from them, along with a multimedia database/encyclopedia of historical information, to create a simulation — a virtual reality.

For example, say you're interested in what's going on in the local school system. You wonder about its effectiveness and want to examine student performance. You enter the mirror world through an "infomachine" and ask how PS-22 is doing. It looks through the performance results for PS-22, other schools in the district, and other districts, browsing historical data to answer your question. You want to see what's going on in your son's classroom right now, and the infomachine brings up a video with background information. Something catches your attention and you want to see last week's history class, so you tune in on that.

You sense some tie-in to the design of the building and turn to a "top-sight" view of the school. From here you zoom down to a more detailed view. You tap historical information on the funding, design, and construction of the building, as well as comments from the architect. You wonder how other parents view the situation, so you contact one — who is wandering around the school through an "infomachine." You could communicate with the principal, teachers, or school board. In short, you can generate a mirror world of the educational system flexible enough to look at any public aspect of it, present or past, in a multimedia format.

Some key elements of mirror worlds include

- the ocean of multimedia information readily available,
- a mechanism for selecting relevant information,
- an "infomachine" to organize the data into a constructive answer to user questions, and
- security systems robust enough to control user access to appropriate authorized information.

Clearly, the elements are not all in place, though Gelernter believes the technology is ready to move in the direction of mirror worlds for every area of interest: government, business, academia, entertainment, the arts — you name it. You can tap in on deliberations, evaluate data, review history, and talk to consultants in any of these areas. It's all there for the asking, sometimes for a price. You can view and use whatever public information you desire and whatever private information authorized to you.

The vision is bold.

On the whole, predictions such as *Mirror Worlds* haven't fared very well when compared with what actually occurs, so don't invest heavily in Mirror Worlds International Ltd. just yet. However, you can expect this book to stimulate your thinking and generate research ideas and goals.

James W. Berkovec
Claremont Graduate School
Claremont, California

Software Construction by Object-Oriented Pictures: Specifying Reactive and Interactive Systems

*George W. Cherry (Thought**Tools, Inc., Canandaigua, N.Y., 1990, ISBN 0-9625003-0-5, 191 pp., $28.95)*

SCOOP-3 (Software Construction by Object-Oriented Pictures of Processes and Parts) is a skillful methodology for producing system and software specifications. The author draws on his experience with Pamela and analysis of other methods to integrate the strengths of graphic notation, object-oriented methodologies, and Ada methodologies effectively in a single specification tool.

The book is well-written with drawings and examples that clearly demonstrate SCOOP-3's effectiveness. In short, this book impressed me.

Several points are worth noting. While the author repeatedly states that SCOOP-3 can be used solely as a specification method, he also stresses its possible employment throughout the software life cycle. In other words, should tools become available, SCOOP-3 specifications can be used to generate Ada code directly. This code can then be compiled and tested (using additional tools) against the software test plans already generated.

Another point: SCOOP-3 is obviously most useful when the implementation language is Ada. This is evident not only from its symbols and notation but most obviously from its reliance on the message as the sole means of interprocess communication. However, such methods as semaphores, monitors, and buffers are also available to SCOOP-3 users through object idioms (or templates).

The text has three parts: a description of SCOOP-3, its comparison with other methods such as structured analysis, and a description of each of its object idioms.

The book could be criticized for treating SCOOP-3 a bit too enthusiastically. For Ada-based systems, this enthusiasm is well-justified. However, SCOOP-3 may not be as useful for other languages and design methods.

It is disappointing that at the time of the text's publication, no CASE tools were available to implement SCOOP-3. Such tools would be extremely useful for the specification of reactive and interactive computer systems. Still, taken as a whole, both this book and the SCOOP-3 specification method are excellent additions to the field.

David W. Mutschler
Naval Air Warfare Center
Warminster, Pennsylvania

"Any clod can have the facts, but having opinions is an art."
Charles McCabe, *San Francisco Chronicle*

THE OPEN CHANNEL

Editor: Will Tracz, IBM, Federal Systems Company, Mail Drop 0210, Owego, NY 13827
e-mail tracz@owgvm0.vnet.ibm.com

Two heads are better than one

The late Sam Rayburn, 17-year speaker of the US House of Representatives, once said something to the effect that when two people agree on everything, one of them is not thinking. That was Mister Sam's pithy way of saying that a second opinion is often valuable.

The late President Kennedy used to *insist* on second opinions in meetings of his Cabinet and the National Security Council. In fact, Kennedy would appoint a devil's advocate for each such meeting, designating someone to argue the "other" side if that became necessary. Furthermore, he always got "morning after" opinions on really crucial issues, because he knew that some people need to ponder questions at length.

In the sciences, especially in the last decade, we have often neglected second opinions. After all, can't we compute *the* answer? Or, if we can't compute it, can we legislate it? For example, Congress tells us that Ada really is the best programming language, in spite of much contrary experience.

As an example of the need for second opinions, I thought a lot after reading the Letter to the Editor and reply (*Computer*, April 1992, p. 4) about R.W. Hamming's review (*Computer*, January 1992, pp. 134-135) of Peter Fejer and Dan Simovici's book, *Mathematical Foundations of Computer Science, Volume 1*.

Ordinarily, I would give all benefits of the doubt to Hamming because of his reputation. But I read — and tested — the code for the ACM Quicksort, and it does indeed do the right thing for a string less than or equal to one. If a book is worth a review in *Computer*, then maybe it is worth two opinions — one from a theorist and another from a practitioner. One can assess the foundational value of the work; the other, its usefulness on the job.

Consider the differences between the reviewer and the authors on the subject of demonstrating program correctness: formal proofs versus well-disciplined testing.

Why not some of each? In all fairness, both sides probably do believe in some of each. Two reviewers of the book might have developed different viewpoints, without either defending any personal feelings.

One benefit of such disagreements is the assurance it gives ordinary mortals to realize that the best and brightest don't have all the answers, either. (I sure don't. And, if I did, I'd be selling them, not writing this.)

The disturbing part is that the field of computer science — now including communications and information systems — is so broad that, unless there is a clear statement of thesis, then antithesis, and eventually synthesis, any of us might be misled.

Reviewers frequently supply the balance. I am not suggesting that we can always reach synthesis — except perhaps to agree to disagree. Examples include our various opinions about the best computer, the best programming language, and the best operating system, as well as the best software development methods and tools.

There are *winners* — with the emphasis on the plurality. Editorials, advertisements, and government decisions notwithstanding, there is no single "best" in any category for all problems.

I believe that the IEEE Computer Society is well qualified to honor such diversity and benefit from the natural competition that follows.

Robert G. Estell
Ridgecrest, California
estell%fidler.decnet@scfb.nwc.navy.mil

Confessions of an Open Channel editor

Am I paranoid, or just mildly deranged? The truth is that sometimes I find it hard to decide what should appear on this page. I am aware that power corrupts and absolute power corrupts absolutely, but I often wonder what electrical power does if you are using alternating current?

As you can see, I am vacillating; but that's just my point. Since I assumed the position of Open Channel editor in March 1991, I have received very few complaints from anyone other than the authors whose contributions I have chosen not to publish. This tacit approval has left me with an insecure feeling that, as ultimate referee, I at times might have thrown the baby out with the bath water (not that I haven't asked certain associates to review some departmental submittals as to the reasonableness of their technical content or lack there of).

Therefore, I appeal to you, the reader, for guidance in the criteria I should apply for selection of what appears before your eyes. To date, my selection criteria have required submittals to be somewhat innovative, provocative, and/or witty. Above all, they have to have some byte/bite (computer relevance and be well written). Up to now, each selection has been a judgment call on my part; in other words, I know it when I see it.

Therefore, I am asking you if you think I need new glasses.

Will Tracz

"The Open Channel" is exactly what the name implies: a forum for the free exchange of technical ideas. Try to hold your contributions to one page maximum in the final magazine format (about 850 words).

We'll accept anything (short of libel or obscenity), so long as it's submitted by a member of the IEEE Computer Society. If it's really bizarre, we may ask you to get another member to cosponsor your contribution.

Send everything to Will Tracz at the address above.

CALL FOR PAPERS

11th IEEE VLSI TEST SYMPOSIUM

Bally's Park Place Casino Hotel
Atlantic City, NJ
April 6 - 8, 1993

ADVISORY COMMITTEE

M. Modi
Naval Air Weapons Center

W. Radcliffe, IBM

N. Kornfield
Widener University

PROGRAM COMMITTEE
(to include)

W. Debany, Vice Chair
USAF Rome Laboratory

J. Monzel, Vice Chair
IBM

K. Cheng, AT&T Bell Labs

B. I. Dervisoglu
Hewlett Packard

J. Ferguson
University of California

H. Fujiwara
Meiji University

J. Hayes
University of Michigan

E. McCluskey
Stanford University

S. F. Midkiff
Virginia Polytechnic Inst.

J. Mucha
University of Hannover

P. Nagvajara
Drexel University

M. Nicolaidis, IMAG/TIM3

I. Pomeranz
University of Iowa

D. E. Ross
Texas A&M University

D. Saab
University of Illinois

J. Savir, IBM

S. C. Seth
University of Nebraska

J. P. Shen
Carnegie Mellon Univ.

A. Singh
Auburn University

M. Soma
University of Washington

P. Varma, Teradyne

D. Wu, IBM

Y. Zorian, AT&T Bell Labs

DESIGN, TEST and APPLICATION: SYSTEMS on SILICON

IEEE VLSI Test Symposium explores state-of-the-art test concepts and trends in VLSI devices designed as microsystems. The eleventh edition of this annual symposium will focus on improved approaches to overcoming test barriers raised by the ever-increasing complexity and lack of test accessibility of today's systems on silicon.

Topics include, but are not limited to, the following:

- Multi-chip Module Test
- Concurrent Checking
- Design for Testability
- Failure Mode Analysis
- Test Generation
- Ultra High Frequency Test
- Delay Test
- Neural Network Test
- Mixed Signal Test
- Test Standards
- Built-in Self-Test
- Simulation
- Test Systems
- ATE Architecture
- Design Verification
- IDDQ Test
- Boundary Scan
- Fault Modeling and Diagnosis
- Quality & Reliability
- Board Test
- System Test

The Program Committee invites authors to submit **seven copies** of a review package consisting of a title page and paper proposal. On the title page include: title, author(s), affiliation(s), suggested topic(s), and an abstract of 50 words or less. **Identify the contact author** and include a complete mailing address, phone number, fax number and e-mail address (if applicable). The paper proposal may be an extended summary or a full paper. **The Program Committee will give preference to full paper submissions.** In either case, clearly describe the nature of the work, explain its significance, highlight novel features, and describe its current status. **Submissions must be original, unpublished material.** Authors of selected papers must prepare a manuscript for the symposium digest. Awards may be given for outstanding papers. The committee invites proposals for panels and tutorial topics.

Submit all papers and proposals to:

Dhiraj Pradhan, Program Chairperson
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112 USA
Tel: (409) 862-2438
Fax: (409) 862-2758
Email: pradhan@ecs.umass.edu

For general information, contact:

Kedong Chao, General Chairperson
Johns Hopkins University APL
Johns Hopkins Road, M/S 23-270
Laurel, MD 20723
Tel: 301 953-6121
Fax: 301 953-6696
Email: kdc@aplcomm.jhuapl.edu

- *Review Package (proposal) due:* **October 9, 1992**
- *Acceptance notification:* November 20, 1992
- *Camera-ready copy due:* **January 29, 1993**

Sponsors: **IEEE Computer Society Test Technology Technical Committee** *and* **IEEE Philadelphia Section**

IEEE COMPUTER SOCIETY

IEEE
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Chicago, Illinois

September 21-25, 1992

COMPSAC 92

McCormick Center Hotel, Chicago, Illinois

The IEEE Computer Society's Sixteenth Annual International

Computer Software & Applications Conference

TUTORIALS: September 21-22, 1992 • CONFERENCE: September 23-25, 1992

TUTORIALS

Monday, September 21, 1992

- Object-Oriented Software Development
  by Suzana Hutz, Motorola, Inc.
- Software Process Improvement: Processes, Methods, Examples
  by Robert Lai, Software Productivity Consortium

Tuesday, September 22, 1992

- Domain-Specific Architectures and Megaprogramming
  by Will Tracz, IBM Federal Systems Company
- Total Software Quality Assurance
  by Tsun Chow, AT&T Bell Laboratories

CONFERENCE-AT-A-GLANCE

Day & Date → / Time ↓ / Rooms Assigned & Tech Areas	Wednesday, September 23, 1992				Thursday, September 24, 1992				Friday, September 25, 1992		
	9:00 - 10:30 a.m.	11:00 a.m. - 12:30 p.m.	2:00 - 3:30 p.m.	4:00 - 5:30 p.m.	9:00 - 10:30 a.m.	11:00 a.m. - 12:30 p.m.	2:00 - 3:30 p.m.	4:00 - 5:30 p.m.	8:30 - 10:00 a.m.	10:15 a.m. - 11:45 a.m.	12:00 - 1:30 p.m.
Room A Parallell & Distributed Network Software	*OPENING SESSION & KEYNOTE ADDRESS* What Is the Reality of Object-Oriented Development? G. L. Tribble, Vice President, NEXT, Inc., USA	Panel: Research & Application Perspect of Formal Methods B. Cheng A-1W	Programming Environments H. Wedde A-2W	Software Metrics & Measurements B. Yacobellis A-3W	*PLENARY PANEL* Software Process Assessment and Improvement S.S. Yau, University of Florida, USA	Distributed Run-Time Support K. Mori A-1T	Fault Tolerance Concepts K. Lin A-2T	Software Tools J. Siddiqi A-3T	*PLENARY PANEL* Privatization of the Internet C.J. TerMaat, Andersen Consulting, USA	Panel: Why Do We Need Extended Transaction Models? R. Rusinkiewicz A-1F	Simulations & Animations L. Ni A-2F
Room B Software Methods, CASE Tools, & Management		Software Process Models R. Lai B-1W	Software Metrics & Models D. Corino B-2W	Panel: Management of Emerging Software Technologies V. Vaishnavi B-3W		Panel: Industrial Software Engineering Training Programs J. Urban B-1T	Specification & Requirement F. Hsia B-2T	Formal Specification O. Shigo B-3T		Testing and Debugging C. Chang B-1F	Computer-Supported Cooperative Work T. Moher B-2F
Room C Object-Oriented & Knowledge-Based Systems		Object-Oriented Methodology T. Ng C-1W	Panel: Open Systems Object Models T. Escamilla C-2W	User Interfaces B. McMillin C-3W		Data & Object Bases B. Wah C-1T	Panel: Integrating Object & Relational Technologies S. Danforth C-2T	Software Applications M. Aoyama C-3T		Expert Systems Y. Matsumoto C-1F	Intelligent Systems W. Lin C-2F

For registration, contact:

David H. Jacobsohn
COMPSAC'92 Registration Chair
5642 South Harper Avenue
Chicago, Illinois 60637, USA
Telephone: (312) 752-4562
FAX: (312) 752-4562
Internet: jacobsohn@depaul.edu

Dick B. Simmons
COMPSAC'92 Program Chair
Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
Telephone: (409) 845-2015
Internet: simmons@cs.tamu.edu

Stephen S. Yau
COMPSAC Steering Committee Chair
Computer and Information Sciences Dept.
301 CSE Building
University of Florida
Gainesville, Florida 32611-2024, USA
Telephone: (904) 392-1211
FAX: (904) 392-1220
Internet: yau@cis.ufl.edu

For additional information, contact:

Bob Yacobellis
COMPSAC '92 Publicity Chair
Motorola, Inc.
3701 E. Algonquin Rd., Suite 601
Rolling Meadows, IL 60008, USA
Telephone: (708) 576-0083
FAX: (708) 576-2025
Internet: rhy@corp.mot.com

Les Shroyer
COMPSAC'92 Conference Chair
Motorola, Inc.
1299 E. Algonquin Rd.
Schaumburg, IL 60196-1077, USA
Telephone: (708) 576-3594
FAX: (708) 576-2259

IEEE Computer Society
Ooshima Building
2-19-1 Minami Aoyama
Minato-ku, Tokyo 107, Japan
Telephone: 81-3-3408-3118
FAX: 81-3-3408-3553
Internet: tokyo.ofc@compmail.com

Sponsored by: IEEE Computer Society ® The Institute of Electrical and Electronics Engineers, Inc.